Commodore & Amiga nr6

czerwiec 1993

MAGAZYN UŻYTKOWNIKÓW KOMPUTERÓW COMMODORE

NR INDEKSU 355216
ISSN 0867-8022



CENA: 12.000 zł

Ray-tracing







- Jak wywołać duchy?
- Europejska TOP-lista - cz.2

Konkurs SUPERSCREEN po raz trzeci

COMMODORE & AMIGA 6/93

AMIGA

C 64

ORAZ



O ray-tracingu prawie wszystko - str.4



Scena - tym razem europejskie hity - str. 10



Dagmara Edytor – świetny edytor sprite'ów w programotece – str. 31

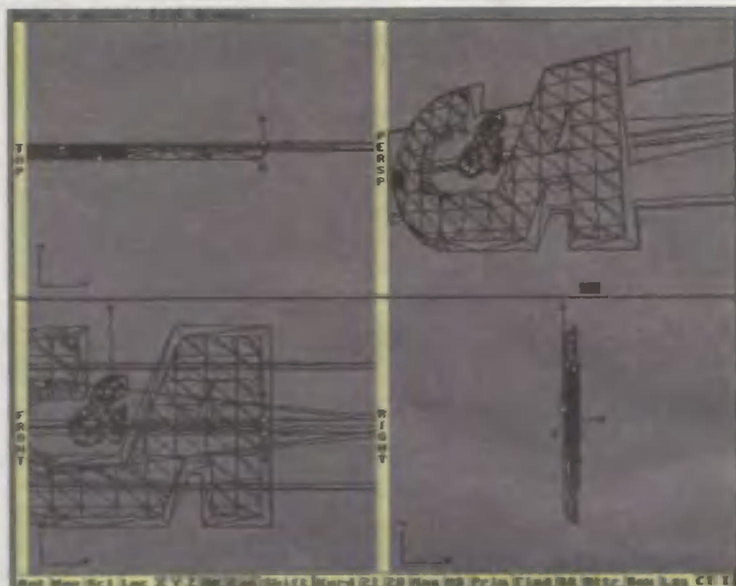
Redaktor naczelny: Klaudiusz Dybowski
Z-ca red. naczelnego: Christian Grzenkowicz
Zespół redakcyjny: Andrzej Bobek (szef działu Amigi), Robert Chojecki, Dariusz Ducki
Opracowanie graficzne: Studio Linea
Zdjęcia: Jerzy Stokowski

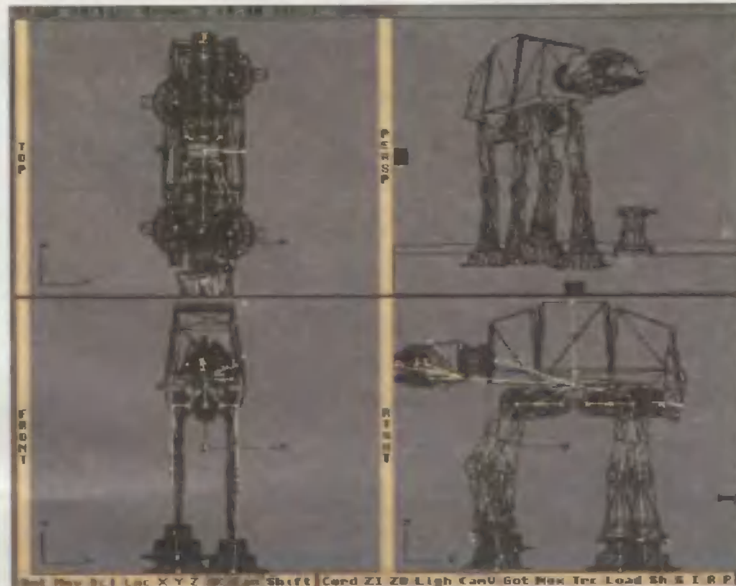
Stali współpracownicy: Rafał Borzyński, Jerzy Dudek, Piotr Cerkiewnik, Bartłomiej Dramczyk, Mariusz Ferdyn, Paweł Galas, Bartłomiej Kachniarz, Marcin Kulesza, Robert Kuliś, Rafał Piasek, Olaf Przybyszewski, Bartosz Smaga, Maciej Strzelbicki
Redakcja: ul. Wasilkowskiego 7, 02-776 Warszawa, tel. 643-18-40
Kontakt z Czytelnikami: piątek w godzinach 13.00-16.00
Dział reklamy: 03-956 Warszawa, ul.Raperswilska 12, tel. 17-50-70 oraz Agencja Reklamowo-Wydawnicza „BYRA" 00-519 Warszawa, ul. Wspólna 41, tel. 625-48-18, tel./fax 29-49-48

Wydawca: Spółdzielnia „Bajtek", ul. Rapelswilska 12, 03-956 Warszawa, tel. 17-50-70
Druk: Przedsiębiorstwo Wydawniczo-Poligraficzne „GRYF", S.A., Ciechanów, ul. Sienkiewicza 51

RAY-TRACING

MAGIA ŚWIATŁA I PRZESTRZENI





Ekran „roboczy" Imagine'a

W ciągu ostatnich kilku lat grafika komputerowa stała się czymś tak oczywistym w telewizji i kinie, że mało kto zwraca już na nią większą uwagę - traktowana jest po prostu jako jeszcze jedna z technik. Można ją spotkać w każdym niemal filmie SF, pełno jest jej też w telewizji.



Na komputerach bowiem tworzy się lwią część wszelkich czołówek, tricków w reklamach, plansz, fruwających po ekranie na najróżniejsze sposoby napisów itd. Nie zdarza się już dziś, nawet w polskiej TV, by w bloku reklamowym nie było choć jednej reklamówki, w której nie użyto by komputerów. Również większość cyklicznych programów swe czołówki zawdzięcza komputerom. Jako zawzięty fan Amigi muszę tu dodać, że ponad połowa tychże reklam i czołówek robiona jest w Polsce na Amigach.

Najtrudniejszą dziedziną grafiki komputerowej, ale równocześnie i najbardziej efektowną, jest animacja trójwymiarowa generowana przez programy używające tytułowej techniki ray-tracingu. Ta dziedzina zastosowań graficznych komputerów jest dominująca i najprawdopodobniej długo jeszcze taką pozostanie. Amiga jest natomiast komputerem coraz bardziej wyspecjalizowanym w dziedzinie tworzenia grafiki, w tym też i animacji trójwymiarowych. W artykule tym chciałbym więc Wam wyjaśnić, czym jest ów magiczny ray-tracing i jak powstaje animacja.

Ray-tracing - co to jest?

Ray-tracing oznacza tyle co „śledzenie biegu promieni świetlnych" - programy używające tej techniki symulują bowiem w pamięci komputera sztuczny, wektorowy świat i tworzą obraz właśnie na podstawie śledzenia „rozchodzących" się w nim promieni światła. Tak to można powiedzieć w jednym zdaniu niewiele z pewnością wyjaśniającym większości z Was. Postaram się więc teraz rozwinąć je do, powiedzmy, kilkudziesięciu.

Podstawy

W stwierdzeniu, że komputer tworzy sobie w pamięci wektorowy świat, nie ma tak wiele przesady, jakby to się mogło wydawać. Twórca animacji musi przecież w jakiś sposób poinformować komputer o wyglądzie obiektów, najprościej zaś zrobić to definiując je w postaci wektorowej siatki. Podstawową jej częścią jest linia prosta (nazywana w programach „edge") łącząca dwa punkty. Podstawową zaś jednostką płaszczyzny jest trójkąt („face") utworzony przez trzy proste. Nie będzie na przykład płaszczyzną prostokąt zbudowany z czterech prostych, jeśli jednak dodamy w nim przekątną, stanie się on dwoma złączonymi trójkątami i w tym dopiero momencie program zacznie uważać go za płaszczyznę. Może się to wydawać mniej lub bardziej udanym rozwiązaniem, a jednak tylko ono zostało zaakceptowane przez użytkowników.

Tak więc cały obiekt, absolutnie każdy detal, musi być utworzony z trójkątów. Byłoby to oczywiście niemożliwe do zrealizowania całkowicie ręcznie, dlatego też programy do animacji trójwymiarowej mają zawsze bardzo rozbudowany moduł edytora obiektów oferujący dziesiątki funkcji ułatwiających tworzenie obiektów.

Dla przykładu, kieliszek tworzymy rysując dwuwymiarową połowę jego przekroju, po czym ją obracamy (tworzymy bryłę obrotową). Wszelkie nierówne płaszczyzny tworzymy zniekształcając przy pomocy odpowiednich funkcji obiekty płaskie. Istnieją oczywiście trudniejsze zadania, jak na przykład stworzenie obiektu z wyciętym w nim otworem o ściśle określonym kształcie (może to być jakiś napis). Zadanie wydawałoby się trudne, jeśli nie niemożliwe do zrealizowa-

nia, gdyby nie rozbudowane możliwości edytorów obiektów programów takich jak Imagine czy Real 3D. Wystarczy w nich bowiem użyć tzw. operacji Booleana (żył kiedyś taki matematyk) a program już sam wykorzysta zadany kształt jako matrycę i wytnie w obiekcie żądaną dziurę. Jak widać, nie taki diabeł straszny...

Marmur czy glina?

Stworzenie siatki wektorowej obiektu to jeszcze nie wszystko, teraz należy bowiem zdefiniować jego wygląd, czyli, mówiąc w najprostszy możliwy sposób, „materiał", z jakiego obiekt jest „zrobiony". Możliwości mamy w tym momencie absolutnie bez liku. Przede wszystkim określa się takie parametry jak kolor, przezroczystość, kolory odbijane, stopień rozpraszania światła odbijanego, stopień gładkości powierzchni. To jednak nie wszystko - trzy pierwsze parametry oraz tzw. bump, czyli miejscowe podwyższenia powierzchni (na przykład monetę robi się nakładając na płaski krążek obraz orła czy reszki jako bump) można definiować przy pomocy nakładania na obiekt grafiki bitmapowej, a także przypisując mu specjalny algorytm definiujący właściwości powierzchni (nazywany teksturą matematyczną).

Z pewnością wiele wyjaśni tu przykład. Jeśli więc chcemy uzyskać marmurową kulę, to nakładamy po prostu na nią grafikę wyobrażającą fragment marmuru. Jeśli chcemy, by kula ta była w niektórych miejscach przezroczysta (mimo, że jest z marmuru!), załatwiamy to nakładając dodatkowo następną grafikę, od której zależeć będzie, w których miejscach kula będzie przezroczysta, a w których nie. Jeśli jeszcze zechcemy, by w jaśniejszych miejscach powierzchnia kuli była lekko uwypuklona, nakładamy grafikę wyobrażającą marmur jako tzw. bump-brush (czyli obraz definiujący bump). Wszystko to jest, jak widać, nieco skomplikowane, możliwości są jednak więcej niż olbrzymie.

I stało się światło...

Gdy gotowe są już obiekty, trzeba je jeszcze jakoś umieścić w przestrzeni. Ta część pracy ma wiele wspólnego z fotografią. Dlaczego? Po prostu tutaj także musimy odpowiednio dobrać oświetlenie, ustawić kamerę itd. Rożnica jest jednak zasadnicza - tutaj mamy absolutnie nieograniczoną swobodę. Ustawiając bowiem lampy, możemy ich zdefiniować tyle, ile nam się spodoba, mogą one mieć dowolne kształty i kolory, oświetlane przez nie obiekty mogą rzucać cienie lub nie. Mało tego, jako lampę można zdefiniować dowolny obiekt, który, jeśli będzie to na przykład marmurowa kula, zacznie się zachowywać jak projektor slajdów, rzucając na całą scenę pokrywający go wzór. To nadal jeszcze nie wszystko. Mamy też możliwość zdefiniowania tzw. światła rozproszonego, czyli oświetlającego wszystko niezależnie od obecności lamp. Na całą scenę można nałożyć także tzw. globalbrush, czego efekt jest mniej więcej taki, jakby całą scenę zamknęło się w wielkiej kuli pokrytej wskazaną przez nas grafiką, która będzie się następnie odbijała w obiektach. Tak na przykład robi się popularny efekt odbijających się w kryształach chmur.

Animacja

Ostatni etap to zdefiniowanie ruchu obiektów. Są dwie podstawowe techniki. Pierwsza polega na tym, że w określonej klatce animacji definiujemy pewne położenie, kąt ustawienia, a także skalę, podobnie w następnej. Teraz program wylicza już tylko animację, w czasie której doprowadza od stanu z pierwszej klatki do stanu z ostatniej. Tak więc obiekt może w tym czasie przelecieć z jednego krańca ekranu na drugi, obrócić się wokół własnej osi, a także płynnie zmienić rozmiary.

Druga technika to dokładne definiowanie toru lotu obiektu, tzw. path. Tor ten może mieć najbardziej wymyślny kształt, może być zapętlony lub nie, zaś obiekt będzie po prostu w trakcie animacji poruszał się wzdłuż niego. Obie techniki mogą być łączone, na przykład obiekt frunący wzdłuż zadanego toru może zmieniać ustawienie i rozmiary. Zresztą w bardziej zaawansowanych programach, jak na przykład Real 3D v2, możliwe jest takie zdefiniowanie ruchu wzdłuż ścieżki (path), że obiekt będzie obracał się w zależności od kierunku ruchu (na przykład samolot, ktoremu latanie bokiem zawsze sprawia trudności), albo też dostosowywał swój kształt do kształtu ścieżki (w ten sposób stworzenie animacji sunącego węża, dość trudnej na Imagine z uwagi na konieczność ręcznej modyfikacji obiektu, jest po prostu śmiesznie proste).



To nie jest skaning ! Wszystkie elementy oraz „nałożenie" światła wykonano za pomocą Imagine'a.

Efekty specjalne

Animację można tworzyć wykorzystując również wszelakiej maści efekty specjalne. Jednym z nich jest tzw. morphing (metamorfoza) jednego obiektu w drugi. Jeśli chcemy na przykład uzyskać znany z „Terminatora 2" efekt pana podnoszącego się z podłogi, wystarczy (w uproszczeniu) przerobić obiekt wyobrażający owego gościa na płaski, a potem zdefiniować animację, w której jedna jego wersja (ta spłaszczona) przemienia się w drugą (normalną) - wszystkie zaś etapy pośrednie wyliczy program. Pamiętajcie jednak, że tak to wygląda w teorii, w praktyce zaś wszystko jest nieporównanie trudniejsze.

Oprócz metamorfozy możliwe są oczywiście wszelkie inne efekty, jak na przykład eksplozja (efekt ten wykorzystano na przykład w zrobionej na Imagine czołówce „Reklama" w polskiej TV).



SŁOWNICZEK

Morphing - efekt płynnej przemiany jednego obiektu w drugi. Można się tu pokusić o rozróżnienie morphingu rastrowego, realizowanego przez takie programy jak Cinemorph, MorphPlus czy ImageMaster, i morphingu obiektów wektorowych w programach do animacji trówymiarowej. O ile w pierwszym przypadku nie ma żadnych ograniczeń, o tyle dostępne dla Amigi programy do ray-tracingu wymagają, by na obiekty docelowy i początkowy składała się taka sama liczba wektorów, co jest niestety znacznym ograniczeniem.

Motion morph - w odróżnieniu od zwykłego morphingu rastrowego chodzi tu nie tylko o zwykłą przemianę dwóch nieruchomych obrazów, ale także o uwzględnienie ruchu. Za przykład może służyć tu przemiana jednego biegnącego człowieka w drugiego.

Ray-tracing - technika wyliczania, na podstawie zdefiniowanych obiektów wektorowych, trójwymiarowych, w pełni symulujących rzeczywistość obrazów. Wykorzystywane tu algorytmy są bardzo skomplikowane i w konsekwencji czasochłonne, uwzględniają one bowiem takie szczegóły jak załamania światła, jego rozproszenie, odbicia, cienie itd.

Tekstura (ang. texture) - w dużym uproszczeniu można powiedzieć, że są to parametry materiału przypisane obiektowi. Te parametry to między innymi kolor, kolory odbijane, kolory przepuszczane (istnieje tu rozdział), współczynnik załamania światła, gładkość powierzchni.

Brush-mapping - definiowanie koloru (lub innych parametrów, takich jak przezroczystość czy też tzw. altitude-mapping, czyli wypukłości powierzchni, co daje efekt płaskorzeźby) obiektu przez nałożenie (nawinięcie) na niego jakiejś grafiki. Za przykład może tu posłużyć walec - gdy nawiniemy na niego grafikę przedstawiającą stylizowany napis Coca-Cola, otrzymamy coś, co przy odrobinie dobrej woli można uznać za puszkę.

Oprogramowanie

Historia profesjonalnych zastosowań graficznych Amigi nie jest wcale tak długa, jakby się to mogło wydawać. Dopiero bowiem ostatnie półtora roku przyniosło sprzęt i oprogramowanie na poziomie profesjonalnym, zaś oprogramowanie naprawdę najwyższej klasy liczy sobie po kilka miesięcy. W dalszej części tego artykułu znajdziecie coś na kształt omówienia rynku oprogramowania do animacji trówymiarowej na Amidze, w którym postaram się przedstawić wszystkie liczące się programy, od tych najprostszych, od których większość ludzi zajmującuch się grafiką na Amidze zaczynała, aż po ukoronowanie w postaci Reala 3D v2, mającego iście rewolucyjne możliwości.

Sculpt Animate 4D

Jego historia liczy już sobie pięć lat. Był on pierwszym w miarę rozbudowanym programem do ray-tracingu na Amidze i to chyba właśnie on zapoczątkował „graficzną" tradycję Amigi.

Ostatnia wersja Sculpta pochodzi sprzed lat, program ma więc, jak na dzisiejsze standardy, mówiąc oględnie, niewielkie możliwości. Jeżeli ktoś go jeszcze używa, to tylko dlatego, że ma on minimalne wymagania sprzętowe - w ograniczonym zakresie może pracować nawet na Amidze 500 z 1 MB RAM-u i bez twardego dysku (żeby było śmieszniej, na takim właśnie zestawie stworzyłem swoją pierwszą komercyjną animację, od której cała przygoda się zaczęła). Sculpt dostępny jest w kilku wersjach, m.in. w wersji "junior" o najmniejszych wymaganiach (i możliwościach).

Turbo Silver

Program niewiele młodszy od Sculpta, mający jednak dużo większe możliwości. Bił go na przykład na głowę w kwestii możliwości definiowania powierzchni obiektów. Był natomiast dużo trudniejszy w obsłudze i o wiele bardziej "pamięciożerny". W przeciwieństwie do Sculpta, Turbo Silver doczekał się jednak następcy. Jakieś dwa lata temu pojawił się bowiem na rynku...

Imagine

W tamtych czasach możliwości Imagine'a były wręcz rewelacyjne. Zmieniono filozofię pracy z programem, który, w zestawieniu ze swym przodkiem, był wręcz fantastycznie łatwy w obsludze. Rozbudowano i tak duże w Turbo Silverze możliwości definiowania powierzchni obiektów na skalę spotykaną w owych czasach tylko w oprogramowaniu Macintosha i "dużych" komputerów. Absolutnie nieporównywalne z jego poprzednikami były też możliwości animacji Imagine'a. Zbudowanie animacji niemożliwej do stworzenia przy użyciu dotychczas dostępnego oprogramowania stało się z chwilą pojawienia Imagine'a kwestią minut. Żaden z poprzednio istniejących programów nie umożliwiał łączenia w animacji wszelkich możliwych technik i, dodatkowo, mieszania ich z wszelakiej maści efektami specjalnymi.

Dość szybko pojawiła się wersja 2.0 programu, będąca do dzisiaj standardem w dziedzinie grafiki trójwymiarowej na Amidze. Problem tylko w tym, że czas mija i to, co było genialne dwa lata temu, dziś jest już tylko przeciętne i najwyższy czas na wypuszczenie nowej wersji. Jest ona zapowiadana już od dłuższego czasu, wciąż jej jednak nie widać, co w zestawieniu ze stwierdzeniami firmy Impulse, twórcy Imagine'a, że nie opłaca jej się rozbudowywać programu ponad miarę, nie nastraja zbyt optymistycznie.

Real 3D

Program ten powstał w Europie i, obok Scali Multimedia, zdaje się być oznaką delikatnego słabnięcia dominującej pozycji Amerykanów w dziedzinie profesjonalnego oprogramowania dla Amigi.

Pierwsze wersje nie były rewelacyjne, nie były nawet świetne. Zdecydowano się bowiem na zupełnie nowy, szaleńczo niewygodny sposób definiowania obiektów, zrażając w ten sposób ludzi do nieźle, mimo wszystko, zapowiadającego się programu. Czas



mijał i program udoskonalano aż do wersji 2.0, którą (jeszcze nie ukończoną) miałem możliwość oglądać kilka miesięcy temu na targach World of Commodore 92 we Frankfurcie. Muszę tu szczerze przyznać, że rzuciła mnie ona niemal na kolana i jeśli Imagine 3.0 nie będzie wręcz fenomenalny, jego szanse w pojedynku z Realem będą mierne. Wersja 2.0 oferuje bowiem funkcje niesłychanie wręcz wyszukane (możliwe dotychczas do znalezienia właściwie tylko w programach tego typu dla graficznych stacji roboczych), takie jak własny język programowania, symulacja zależności ostrości obrazu obiektów od ich oddalenia od kamery, czy też uwzględnienie grawitacji i sprężystości przy wyliczaniu ich ruchu (za przykład może tu służyć animacja kul na stole bilardowym, którą tworzymy nadając jedynie pewną prędkość białej bili; funkcja ta była dotychczas absolutnie niespotykana w programach w cenie poniżej 20000 $). Real 3D v2 oferuje też takie niespotykane w programach konkurencyjnych funkcje jak generacja fraktalowych krajobrazów, możliwość użycia animacji zamiast statycznej grafiki przy brush-mappingu, zdolność do dokonania morphingu dwóch użytych przy brush-mappingu grafik, pracę z wektorowymi czcionkami AGFA Compugraphic, a wreszcie możliwość używania krzywych wektorowych.

LightWave 3D

Niecałe dwa lata temu w USA pojawiła się karta graficzna dla Amigi o nazwie Video Toaster. Oprócz wielkich możliwości sprzętowych karta była jeszcze wyposażona w doskonałe oprogramowanie, któremu z pewnością zawdzięcza w znacznej mierze oszałamiający sukces w USA. Lightwave 3D jest dodawanym do Video Toastera programem do animacji trójwymiarowej. Aktualnie dostępna jest jego wersja 2.0 (niebawem ma się ukazać wersja 3.0), której możliwości są bardzo imponujące.

Po pierwsze Lightwave 3D jest bardzo szybki, a to dzięki zaimplementowaniu podobnych jak w „pecetowym" 3D Studio, uproszczonych algorytmów, nie uwzględniających na przykład załamania promieni świetlnych przy przejściu przez obiekt. Nie jest to jednak, jak w przypadku 3D Studio, dyskwalifikująca często przy poważniejszych zastosowaniach wada, w dostępnej już od dawna wersji 2.0 umieszczono także możliwość pełnego ray-tracingu.

Lightwave 3D jest ogólnie programem o dość dużych możliwościach przewyższającym nieco Imagine'a 2.0 (na przykład bardzo rozbudowanymi możliwościami morphingu obiektów), ustępującym jednak zdecydowanie wersji 2.0 Reala 3D.

Lightwave 3D ma jednak bardzo dużą wadę - do jego uruchomienia niezbędna jest karta Video-Toaster, a ona zaś jest z kolei dostępna tylko w wersji NTSC i najprawdopodobniej nigdy już nie powstanie jej wersja europejska, pracująca w systemie PAL.

Inne

Pięć omówionych powyżej programów jest najbardziej znanych, dla Amigi powstało jednak jeszcze wiele innych programów do animacji trójwymiarowej. Niektóre z nich mają dość duże możliwości, czasem nawet szkoda, że nie zyskały większej popularności. Tak więc grupę tę tworzą: Caligari (w uproszczonej wersji dodawany do karty graficznej Impact Vision, ostatnio dostępny też jako samodzielny, drogi program Caligari24), Aladdin (dawniej Draw4D), Journeyman, 3DPro, Reflections (program niemiecki, znany szerzej dzięki sławie używającego go Tobiasa Richtera).

Zakończenie

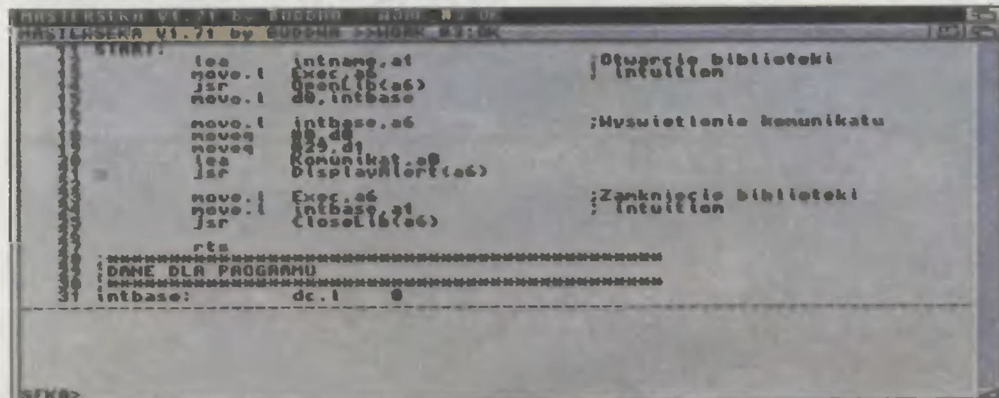
Amiga to komputer wręcz idealny do grafiki i nie bez powodu używa się jej w większości studiów telewizyjnych w naszym kraju. Dzięki najnowszemu oprogramowaniu różnica w jakości grafiki tworzonej na Amidze i dużych maszynach (np. Silicon Graphics) jest wręcz niezauważalna (mniej więcej taka, jak między HAM8 a prawdziwą grafiką 24-bitową).

Mam więc nadzieję, że artykuł ten przybliżył Wam nieco istotę zadania, dla którego Wasz komputer tak dobrze się nadaje.

ANDRZEJ BOBEK

MasterSeka v1.71

Aby napisać jakikolwiek program w języku maszynowym niezbędny jest asembler. Obecnie najpopularniejszymi asemblerami wsród koderów idących z duchem czasu są Trasch-Assembler oraz ASM-One. MasterSeka nie wytrzymuje ich konkurencji, jednak postanowiłem go opisać ze względu na jego ogólnodostępność. Zresztą powiedzmy sobie szczerze - nie jest on taki zły.



W najnowszej wersji programu (1.80) opcje można wybierać z menu, jednak program dosyć często się zawiesza, tak więc do pracy proponuję sprawdzoną wersję 1.71. W tej wersji wyboru opcji dokonujemy przy użyciu klawiatury.

Konfigurowanie MasterSeki

Już w chwili wywoływania MasterSeki z poziomu AmigaDOS-u możemy wpłynąć na konfigurację programu. Dostępne są m.in. następujące opcje (wydajemy je w chwili wywołania programu poprzedzając znakiem „-"):

-b *parametr* (parametr może tu przybierać wartości 0, 1, 2).
Podanie wartości „1" powoduje, iż gadżet przesuwu tekstu w oknie edytora będzie umieszczony z prawej strony, podanie wartości „2" umieszcza ów gadżet z lewej strony, natomiast podanie wartości „0" wyeliminuje go z ekranu.

-c *ilość pamięci*
Program przy uruchomieniu zarezerwuje w pamięci typu CHIP podaną ilość (kilobajty) pamięci.

-e *ilość linii*
Określa ilość widocznych w oknie edytora.

-l *parametr*
Jeżeli jako parametr podamy wartość „0", to zostanie wyłączona numeracja linii w edytorze („1" - włącza numerację).

-r *parametr*
Jeśli parametr przyjmie wartość „1", to wgrywaniu i zgrywaniu będzie towarzyszył FileRequester z biblioteki ARP, zaś jeśli parametr przyjmie wartość „2", będzie to FileRequester z biblioteki REQ (oczywiście biblioteki te muszą znajdować się na dysku). „0" - brak jakiegokolwiek requestera.

-s *parametr*
Jeżeli parametr przyjmie wartość „0", to program uruchomi się w oknie, jeśli będzie to np. „2", to program uruchomi się na nowym ekranie, z liczbą bitplanów równą parametrowi (w naszym przypadku będą to 2 bitplany).

-w X Y XX YY
X,Y - współrzędne lewego górnego punktu okna, w którym uruchomi się program, natomiast XX,YY określają rozmiar okna.

-X
Po uruchomieniu programu zostanie zamknięty ekran Workbencha.

Po uruchomieniu MasterSeki podajemy (jeżeli nie zrobiliśmy tego wcześniej), z jakiego typu pamięci program ma korzystać oraz definiujemy jej wielkość. Po podaniu tych parametrów przechodzimy do okna wydawania komend. Do okna edytora przechodzimy naciskając klawisz ESCAPE (Esc). Nie muszę chyba mówić, iż właśnie w tym oknie piszemy program (przykładowy, napisany w Seka-asemblerze, przedstawia zamieszczony listing).

Funkcje w oknie edycji

OPERACJE NA CIĄGACH ZNAKÓW

AMIGA-S - wpisanie ciągu znaków do przeszukania w kodzie źródłowym, szukanie następuje od pozycji, w której znajduje się kursor,
AMIGA-s lub CTRL-J - szukanie następnego ciągu znaku,
AMIGA-a - szukanie poprzedniego cigu znaków,
AMIGA-R - szukanie i zastępowanie ciągu znaków innym (wprowadzonym) ciągiem znaków,
AMIGA-r - zastępowanie następnego ciągu znaków.

OPERACJE NA BLOKACH TEKSTU

AMIGA-b lub CTRL-B - zaznaczenie bloku tekstu,
AMIGA-c lub CTRL-C - kopiowanie bloku tekstu do bufora,
AMIGA-i lub CTRL-F - wstawienie bloku tekstu,
AMIGA-x - skasowanie bloku oraz przeniesienie go do bufora,
CTRL-U - zamiana na duże litery,
CTRL-L - zamiana na małe litery,
CTRL-R - zamienia blok zaznaczony z buforem,
CTRL-W - zapis zaznaczonego bloku na dysku.

FUNKCJE PRZEMIESZCZANIA

AMIGA-y, AMIGA-x lub CTRL-D, CTRL-K - kasowanie jednej linii,
CTRL-A - poprzedni ekran do wyświetlenia,
CTRL-Y - następny ekran do wyświetlenia,
CTRL-S - przeskok o 100 linii w górę,
CTRL-X - przeskok o 100 linii w dół,
CTRL-T - idź na początek kodu źródłowego,
CTRL-Q - kursor na początek linii,
CTRL-E - kursor na koniec linii,
CTRL-O - wstawienie linii.

INNE FUNKCJE

AMIGA-A - tzw. szybka asemblacja, bez dodatkowych opcji,
AMIGA-Q - przeprowadza asemblację oraz uruchamia napisany program.

Funkcje w oknie komend

W oknie wydawania komend dysponujemy wieloma opcjami, oto ważniejsze z nich:

@komenda - wykonanie komendy CLI, np. DIR,
A - asemblacja,
Badress - naliczenie sumy kontrolnej dla bootblocku,
C - porównywanie ze sobą dwóch obszarów pamięci,
CLS - czyszczenie ekranu,
CL lub KS - kasowanie kodu źródłowego,
D - disassemblacja,
F - szukanie w pamięci ciągu bajtów,
FI - wypełnienie zadanego obszaru zadaną wartością,
J - skok do programu,
KF - kasowanie pliku,
Q - wyświetlenie zawartości pamięci (szesnastkowo),
M - modyfikacja zawartości pamięci,
N - wyświetlenie zawartości pamięci (ASCII),
O - odzyskanie kodu źródłowego po przypadkowym skasowaniu komendą ks,
R - załadowanie kodu źródłowego,
W - zapisanie kodu źródłowego,
RO - załadowanie pliku wykonalnego,
RI - załadowanie danych,
RS - odczyt sektorów z dysku,
W - zapis kodu źródłówego,
WI - zapis danych,
WO - zapis pliku wykonalnego,
WS - zapis sektorów,
U - skasowanie z pamięci pliku wykonalnego,
V - zmiana katalogu (CD) oraz wyświetlenie katalogu dysku (DIR),
AMIGA-q - wyjście lub restart programu.

I to by było na tyle. Nie opisałem tu wszystkich funkcji programu (od tego jest instrukcja obsługi) a jedynie te, z których sam najczęściej korzystam. Niektórzy mogą jednak zadać pytanie: po co tak szczegółowy opis programu? Otóż po pierwsze MasterSeka często (czytaj zawsze) rozprowadzany jest bez żadnej dokumentacji na dyskietkach z innymi programami, a po drugie, jeżeli taka dokumentacja już istnieje, to jest ona z reguły w języku angielskim lub niemieckim.

Na podstawie dokumentacji programu i własnych doświadczeń opracował

MARIUSZ FERDYN

MasterSeka V1.71
Autor: BUDHA of Spreadpoint

Przykładowy listing programu pisanego w asemblerze MasterSeka

```
;**********************************************
;Deklaracje stałych
;**********************************************
        Exec            =       4
        OpenLib         =       -408
        CloseLib        =       -414
        DisplayAlert    =       -90
;**********************************************
;Tu zaczyna się program
;**********************************************
START:
        lea     intname,a1              ;Otwarcie biblioteki
        move.l  Exec,a6                 ;intuition
        jsr     OpenLib(a6)
        move.l  d0,intbase

        move.l  intbase,a6              ;Wyświetlenie komunikatu
        moveq   #0,d0
        moveq   #29,d1
        lea     Komunikat,a0
        jsr     DisplayAlert(a6)

        move.l  Exec,a6                 ;Zamknięcie biblioteki
        move.l  intbase,a1              ;intuition
        jsr     CloseLib(a6)
        rts

;**********************************************
;DANE DLA PROGRAMU
;**********************************************
intbase:        dc.l    0
intname:        dc.b    'intuition.library',0
Komunikat:      dc.b    0,195,11
                dc.b    'NIE PANIKUJ - TO NIE GURU !!!'
                dc.b    0,1
                dc.b    0,240,21
                dc.b    'Commodore & Amiga'
                dc.b    0,0
```

Słownik angielsko-polski

Ostatnimi czasy polski rynek oprogramowania obrodził w różnego rodzaju słowniki języków obcych. Najwięcej jest właśnie angielsko-polskich. Firmy dystrybucyjne konkurują ze sobą i chyba przyniesie to pozytywne efekty. W tym artykule opisany jest słownik angielsko-polski, którego dystrybutorem jest Fundacja Edukacji Technologicznej.

Zawartość pakietu

Dwie dyskietki 3,5-calowe wraz z krótką instrukcją zapakowane są w bardzo praktycznym etui (można je np. postawić między książkami na półce, gdzie równie dobrze się prezentuje). Instrukcja obsługi jest bardzo skromna (cztery strony formatu A5) i zawiera podstawowe informacje o programie.

Co znajduje się na dyskietkach

Na pierwszej dyskietce znajduje się Edytor Wyrazów, Tłumacz, Odpytywacz oraz pełna dokumentacja do programu. Druga dyskietka zawiera dane z tłumaczeniami. Przed uruchomieniem programu, aby móc w pełni wykorzystać jego możliwości, należy zapoznać się z dokumentacją (pierwsza dyskietka).

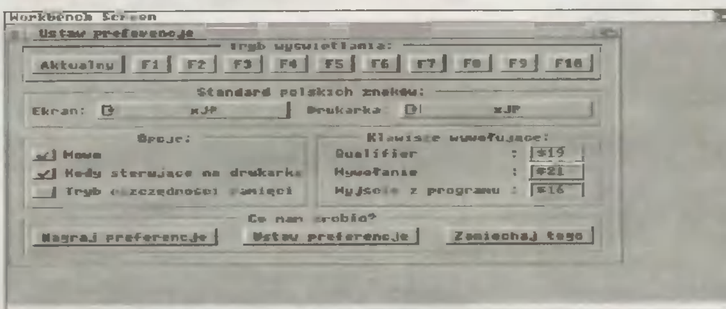
Pierwsze wrażenia

Po zapoznaniu się z instrukcją przeszedłem do testowania programów zawartych na dyskietce. Pierwsze wrażenie nie było pozytywne - podczas pracy z systemem 2.0 (takim właśnie dysponuję) ikony zwyczajnie nachodzą na siebie. No cóż, autor zapomniał o tym drobnym fakcie, inna rzecz, że każdy może sobie poukładać ikony jak chce i zgrać taką konfigurację.

Następnie postanowiłem zapoznać się z dokładnymi instrukcjami zawartymi na dyskietce. I tu spotkał mnie drugi zawód. Otóż autor zastosował błędny „PPMORE" programu do wyświetlania instrukcji zawartych na dysku, wskutek czego jeden z gadżetów w programie nie jest wykorzystany i „kliknięcie" na nim (przynajmniej pod systemem 2.0) kończy się zwykle komunikatem „SOFTWARE FAILURE".

Same instrukcje napisane zostały na szóstkę z plusem. Po przebrnięciu przez nie (około 70 KB tekstu) przeszedłem do testowania poszczególnych programów.

Tłumacz 2.0



Jest to właściwy program tłumaczący. Przed jego uruchomieniem należy zdecydować się na typ czcionki i standard polskich liter jakich będziemy używać. Na dyskietce są trzy rodzaje czcionek, a mianowicie TOPAZ, PEARL i RUBYX, natomiast standardów polskich liter jest dziesięć: xJP, AmigaPL, Mazovia, Latin 2, CSK, DHN, Microvex, Ventura. Możemy także wybrać opcje bez polskich liter, wówczas polskie znaki nie będą wyświetlane a zamiast nich wyświetlane będą ich odpowiedniki. Np. zamiast „ą" będzie wyświetlane „a". W tym miejscu należą się wielkie brawa dla autora, który przystosował program do tak wielu standardów.

Zaraz po uruchomieniu programu musimy przebrnąć przez zabezpieczenie. Polega ono na wpisaniu odpowiedniego wyrazu z oryginalnej instrukcji dołączonej do programu. Zaletą tego rodzaju zabezpieczenia, iż możemy dowolnie kopiować program (choć instrukcję także można powielić). Osobiście bardzo mi się to zabezpieczenie nie spodobało, ponieważ szukanie 6 wyrazu (albo i 16) w 19 linii na stronie „n-tej" nie należy do rzeczy przyjemnych. Autor mógłby się ograniczyć np. do trzech pierwszych wyrazów i pięciu pierwszych linii.

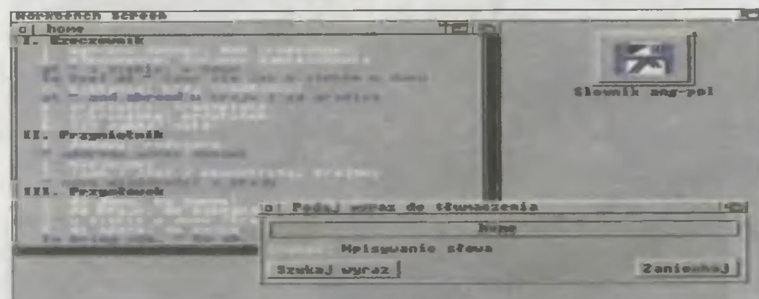
Po „odbezpieczeniu" program instaluje się rezydentnie i możemy go wywołać w dowolnej chwili naciskając odpowiednią kombinację klawiszy. Po wywołaniu ukazuje się okno, do którego wpisujemy żądany wyraz. Może się zdarzyć, że nie znamy całego wyrazu. Wówczas można zastąpić nieznaną część wyrazu znakami specjalnymi. Jeśli tylko jeden wyraz będzie odpowiadał podanemu wzorcowi to automatycznie pojawi się jego opis. Jeśli wyrazów zgodnych z wzorcem będzie więcej, to pojawi się okno ze wszystkimi wyrazami odpowiadającymi wzorcowi.

Obsługa okna ze spisem jest standardowa: spis przewijamy scrollbarem, strzałkami lub klawiszami kursora (góra lub dół) a wyraz do tłumaczenia wybiera się poprzez dwukrotne naciśnięcie lewego przycisku myszy. Wyboru słowa można także zaniechać. Na raz możemy mieć dowolną liczbę okien z opisami wyrazów (limitem jest tylko dostępna pamięć, należy też pamiętać, że im większa ilość okien, tym wolniejsze są operacje graficzne z nimi związane).

Jeśli wyraz nie zostanie znaleziony i nie będzie zawierał żadnej znanej końcówki lub przedrostka, można go poprawić lub zaniechać operacji.

Jeśli wyraz nie zostanie znaleziony, ale będzie

zawierał jakieś znane końcówki np. -ing, -ed, -er, -s itp., wyświetli się okno, w którym w pierwszej części będą wyświetlane znalezione końcówki lub przedrostki, a obok - wyraz po usunięciu tej końcówki lub tego przedrostka. Program jednocześnie sprawdza, czy taki wyraz (już z obciętą końcówką) znajduje się w słowniku. Wyrazy istniejące będą wyświetlane czcionką pogrubioną. Wyraz do dalszego tłumaczenia



możemy wybrać klawiszami kursora i nacisnąć ENTER lub szybko nacisnąć dwa razy lewy przycisk myszy. Zamykając okno lub naciskając klawisz ESC możemy zaniechać wyboru. Funkcja ta jest chyba najwspanialszą rzeczą, jaką można spotkać w tego typu programach.

Jeżeli program tłumacza wywołamy np. podczas pracy w edytorze CED, wówczas komputer wyświetli ekran Workbencha, a po przeprowadzonym tłumaczeniu wróci do edytora CED. Podczas tłumaczenia możemy także wydrukować przetłumaczony wyraz, zmusić Amigę do wypowiedzenia szukanego wyrazu oraz do zmiany preferencji.

Program „Tłumacz" wyposażony jest także w interfejs Arexxa. Na dysku z pakietu znajduje się program w Arexx-ie, który można wykorzystać z poziomu Cygnus Edytora. Służy on do tłumaczenia słowa, na którym aktualnie znajduje się kursor. Aby można było korzystać z tego programu, należy spreparować odpowiednio dyskietkę, co jest dokładnie opisane w instrukcji dyskietkowej.

Pod koniec testu wykryłem pewną wadę programu. Otóż jeżeli wywołamy program z opcją wykorzystania odpowiedniej czcionki, której jednak nie ma na dysku, program wyświetla komunikat „SOFTWARE FAILURE". Jeżeli naciśniemy którykolwiek przycisk myszy, komputer zawiesi się zupełnie.

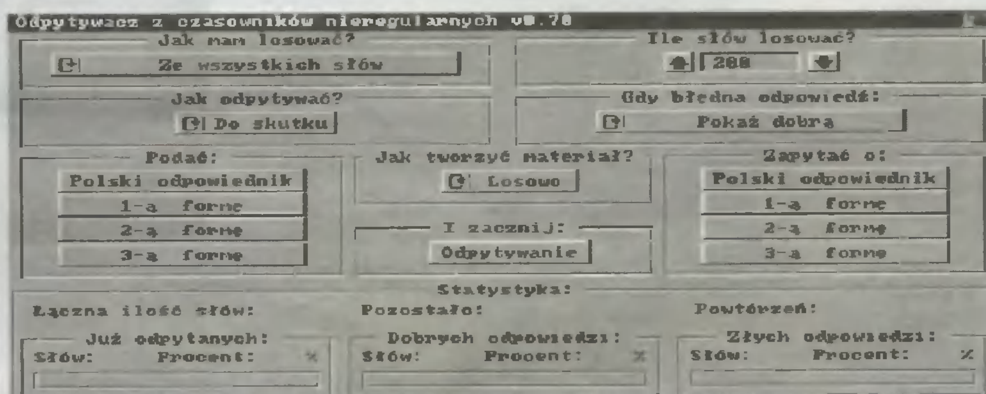
Edytor wyrazów 2.0

„Edytor wyrazów" służy do własnoręcznego dopisywania wyrazów do słownika. Tak jak w przypadku programu „Tłumacz", „Edytor" jest bardzo rozbudowany. Uruchamia się on na własnym ekranie i za jego pomocą możemy dopisywać, kasować i poprawiać opisy wyrazów znajdujących się w słowniku.

Odpytywacz 0.70

Zadaniem tego programu jest odpytywanie z odmiany czasowników nieregularnych. W programie można ustalić ilość odpytywanych czasowników, o którą formę komputer ma zapytać oraz którą formę mamy podać. Komputer wyświetla aktualnie informację, z ilu czasowników już nas przepytano oraz na ile udzielono prawidłowych odpowiedzi.

Także i ten program jest perfekcyjnie dopracowany pod względem funkcjonalnym i graficznym, brakuje może tylko dobrych efektów dźwiękowych. W pakiecie, który otrzymała redakcja, dostarczona była wersja 0.70 „Odpytywacza", co oznacza, że program nie jest jeszcze całkowicie



ukończony. Ale niedoróbek nie ma wcale tak dużo. Zauważyłem tylko, że nie działa ręczne wybieranie czasowników (trzeba się przez to zdać na wybór losowy) oraz pozycja z menu „Informacja gramatyczna" (po informację jesteśmy odsyłani do instrukcji).

Podsumowanie

Opisywany pakiet, mimo kilku mało istotnych wad łatwych do usunięcia, jest produktem jak najbardziej wartościowym. Polecam go szczególnie tym, którzy mają w dostateczny sposób opanowane podstawy języka angielskiego i zajmują się tłumaczeniem tekstów. Pakiet zostanie wykorzystany w pełni, jeżeli będziemy tłumaczyć korzystając z edytora Cygnus - wbudowany interfejs Arexxa pozwala na wskazanie kursorem wyrazu do przetłumaczenia. „Słownik angielsko-polski" mogę także polecić wszystkim uczącym się języka i to nie tylko ze względu na wbudowany odpytywacz, przy pomocy którego możemy opanować odmianę czasowników nieregularnych. Podczas nauki często natykamy się na nieznane słowa a wtedy nie zawsze chce nam się wertować „analogowy" słownik - w takich przypadkach „Słownik..." jest niezastąpiony. Dotyczy to również instrukcji do programów typu Shareware, Freeware, Public Domain, które bardzo często zapisane są na dyskietce wraz z programem - wystarczy kliknąć myszą i nieznane słowo staje się znane.

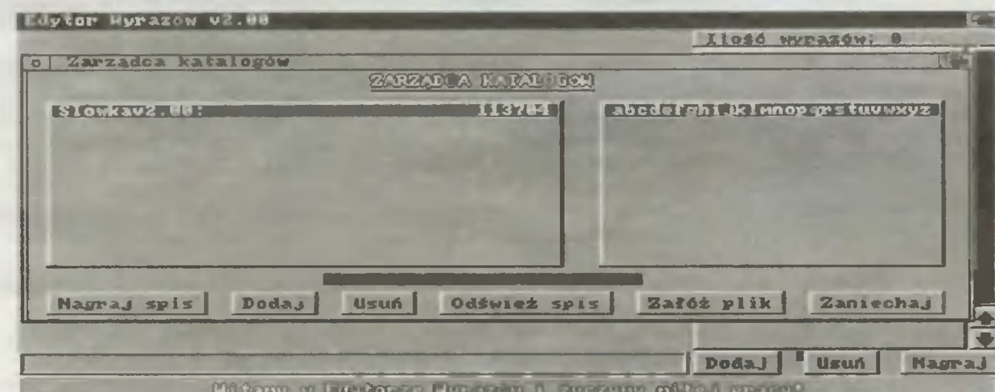
Na koniec wypada więc pogratulować autorowi Krzysztofowi Kowalczykowi za napisanie wspaniałego programu i mieć nadzieję, iż włożona praca jak najbardziej mu się zwróci na niezbyt normalnym polskim rynku oprogramowania. Ja natomiast z niecierpliwością czekam na podobny słownik „w drugą stronę", tzn. polsko-angielski.

MARIUSZ FERDYN

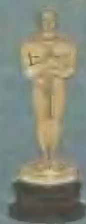
„Słownik angielsko-polski", 2 dyskietki + instrukcja
AUTOR: Krzysztof Kowalczyk
DYSTRYBUTOR: Fundacja Edukacji Technologicznej, 03-480 Warszawa, ul. Burdzińskiego 5, tel. 180176

ZALETY:
+ bardzo dobra instrukcja obsługi (na dyskietce)
+ wygodna obsługa, zwłaszcza przy korzystaniu z edytora Cygnus Editor
+ wbudowany interfejs portu AREXXa
+ duży zasób słów (21 tysięcy)
+ obsługa za pomocą myszy lub klawiatury

WADY:
- niedopracowane rozstawienie ikon pod systemem 2.0
- zastosowanie błędnego „PPMORE" zawieszającego się pod systemem 2.0
- skromna instrukcja obsługi (na papierze)



THE BEST OF



CZYLI EUROPEJSKA TOP-LISTA AMIGOWSKIEJ DEMOSCENY

W tym miesiącu nadeszła wreszcie pora na obiecaną europejską top-listę (konkretnie jej pierwszą część, gdyż jest tego tak dużo, że musiałem podzielić całość na dwa odcinki) najlepszych na scenie. Oparta jest ona na typach Eurocharts nr 18 uznawanych praktycznie przez całą scenę, choć oczywiście zawsze znajdzie się kilku ludzi zarzucających Eurocharts różne manipulacje (podobnie jak polskiemu ZZ TOP).

Ale przejdźmy do rzeczy, czyli do samej listy. Podobnie jak w przypadku polskiej listy, pierwsza liczba oznacza liczbę głosów, druga zdobyte punkty w procentach, trzecia natomiast to poprzedni ranking - z Eurocharts 17.

NAJPOPULARNIEJSZE GRUPY

1. Anarchy	311	19,55 %	(2)
2. Silents	307	19,30 %	(1)
3. Melon Design / Crystal	166	10,43 %	(3)
4. Andromeda	114	7,17 %	(4)
5. Kefrens	95	5,97 %	(5)
6. Sanity	58	3,65 %	(7)
7. Tristar & Red Sector Inc.	43	2,70 %	(11)
8. Phenomena	38	2,39 %	(6)
9. Alcatraz	33	2,07 %	(8)
10. Dual Crew	29	1,82 %	(NEW)
11. Complex	26	1,63 %	(10)
Rebels	26	1,63 %	(8)
Spaceballs	26	1,63 %	(16)
14. Crionics	25	1,57 %	(NEW)
15. Pure Metal Coders	24	1,51 %	(13)
Skid Row	24	1,51 %	(19)
17. Mexx	20	1,26 %	(NEW)
18. Scoopex	15	0,94 %	(11)
19. LSD	14	0,88 %	(16)
20. Reflect	13	0,82 %	(19)

Jak widać, od dłuższego czasu na świecie są w stanie konkurować ze sobą jedynie dwie grupy. W liczbie wypuszczanych produkcji biją one na głowę całą konkurencję (co najmniej pięć demosów całodyskowych rocznie). Najśmieszniejszy w tym wszystkim jest fakt, że żadne z tych demek (za wyjątkiem „Hardwired") nie jest zaliczane do rewelacji. Są to jednak „Firmy" (przez duże F), na których można polegać. Pozostałe grupy z pierwszej piątki to:

MELON DESIGN - podgrupa w ogromnym CRYSTAL, która sformowała się z dawnych memberów SILENTS, ANARCHY, ALIANCE DESIGN/QUARTEX i stworzyła nowy oryginalny styl demosów.

ANDROMEDA - nowa grupa, która zrobiła ogromną karierę w 1992 roku dzięki mistrzostwu designu.

KEFRENS - zreaktywowany po dosyć długiej przerwie KEFRENS jest ponoć lepszy niż kiedykolwiek. Udowodnił to wygrywając compo na Hurricane party trackmem „Guardian Dragon II", ale czy to wystarczy?



Wspaniała twórczość grafika z grupy RED SECTOR

Tak popularne w Polsce SANITY niestety nie załapało się do pierwszej piątki a PHENOMENA, słynna ze swej „Enigmy", spada na łeb, na szyję...

NAJLEPSZE TRACKMA

1. Silents & Crionics - „Hardwired"	334	22,97 %	(1)
2. Kefrens - „Guardian Dragon II"	133	9,15 %	(2)
3. Andromeda - „D.O.S"	91	6,26 %	(3)
4. Melon Design - „Human Target"	83	5,71 %	(5)
5. Alcatraz - „Odyssey"	77	5,30 %	(4)
6. TRSI - „Wicked Sensation"	66	4,54 %	(NEW)
7. Reflect - „Sound Vision"	63	4,33 %	(7)
8. Anarchy - „In The Kitchen"	57	3,92 %	(6)
9. Anarchy - „Flower Power"	42	2,89 %	(NEW)
10. Razor 1911 - „Voyage"	39	2,68 %	(11)
11. Spaceballs - „Wayfarer"	36	2,48 %	(8)
12. Silents - „Static Chaos"	32	2,20 %	(16)
13. Phenomena - „Enigma"	31	2,13 %	(9)
14. Anarchy - „3D Demo"	27	1,86 %	(13)
15. Melon Design - „S.O.S"	26	1,79 %	(9)
16. Silents - „Xpose"	21	1,44 %	(16)
17. Scoopex - „Mental Hangover"	20	1,38 %	(15)
18. Silents - „Enjoy The Silents"	18	1,24 %	(NEW)
19. Sanity - „World Of Commodore"	17	1,17 %	(NEW)
20. Complex - „Delirium"	16	1,10 %	(12)

„Hardwired" niezmiennie na topie! Przez cały rok nie znalazł się dlań godny konkurent. Czy nowe demo grupy SPACEBALLS tego dokona? Na razie jednak „Hardwired" prowadzi z druzgoczącą przewagą, a na dalszych miejscach piąta, czwarta i trzecia grupa świata. Czy to nie dziwne, że w odwrotnej kolejności?

1. Quartex - „Substance"	205	16,87 %	(1)
2. Sanity - „Elysium"	137	11,28 %	(2)
3. Anarchy - „Hardcore"	116	9,55 %	(3)
4. Phenomena - „Joyride"	98	8,07 %	(4)
5. Complex - „Gospel Karaoke"	92	7,57 %	(5)
6. Sanity - „Optimum Fuckup"	[redacted]	5,10 %	(6)
7. Coma - „Hot Dots"	58	4,77 %	(8)
[redacted] Desire - „Menace"	51	4,20 %	(7)
[redacted] Design - „Bomb"	32	2,63 %	(13)
10. Vision - „Can't Be"	28	2,30 %	(10)
11. [redacted] Sector Inc. - „Cebit '90"	26	2,14 %	(11)
12. Shining - „Vector Exterminator"	25	2,06 %	(18)
13. Mexx - „BBS Intro"	19	1,56 %	(NEW)
Cryptoburners - „The Hunt..."	19	1,56 %	(9)
15. Red [redacted] Inc. - „Misery Dentro"	16	1,32 %	(17)
Scoopex - „Seen Before"	16	1,32 %	(NEW)
17. Complex - „Vector Preview"	15	1,23 %	(13)
18. [redacted] - „Sledge Hammer Intro"	13	1,07 %	(NEW)
Dual Crew - „Morpho"	13	1,07 %	(NEW)
[redacted] Silents - „Mentro? Dentro?"	12	0,99 %	(13)
[redacted] - „Quackbusted"	12	0,99 %	(12)
Spreadpoint - „Cube-O-Matic"	12	0,99 %	(16)

Ta rubryka jest niemalże legendarna: od dwóch [redacted] pierwsze dwa miejsca nie uległy zmianie! Naprawdę niesamowite!

Na dalszych miejscach najlepsze demko DANa z ANARCHY i ostatnie demko PHENOMENY (ciekawe, czy PHENOMENA odejdzie bezpowrotnie [redacted] zapomnienie).

NAJLEPSZE DISCMA[redacted]

1. R.A.W. / Pure [redacted] Coders	462	31,26 %	(1)
2. [redacted] Data / Anarchy	173	11,71 %	(3)
3. McDisk / [redacted]	172	11,64 %	(2)
4. Eternal / [redacted]	136	9,20 %	(6)
5. Top Secret / Majic [redacted]	98	6,63 %	(5)
6. I.C.E. / I.C.E Team	70	4,74 %	(4)
7. Grapevine / [redacted]	59	3,99 %	(7)
8. Magbox / Balance	55	3,72 %	(14)
9. Zine / Brainstorm	33	2,23 %	(8)
10. Hack-Mag / D-Tect	31	2,10 %	(9)
11. [redacted] / Mirage	25	1,69 %	(12)
12. Maggy / Complex	23	1,56 %	(10)
13. Nordic Report / Noxious	20	1,35 %	(13)
14. Playbyte / Shining	17	1,15 %	(NEW)
15. [redacted] Rites / Destiny	16	1,08 %	(11)



[redacted] Sensation/TRSI

Dawniej bezsprzecznie najlepszym DiscMagiem na świecie był ZINE, niestety zaprzestano jego wydawania już prawie półtora roku temu. Szkoda, bo część ludzi [redacted] sceny nadal nie może się z tym pogodzić i głosuje na niego (dziewiąte miejsce).

Wróćmy jednak do czasów obecnych. Widać, że pałeczkę przejął R.A.W., uważając się [redacted] spadkobiercę ZINE'a. Nie jest [redacted] jednak to samo, mimo że [redacted] skład redakcji R.A.W.-a wchodzi kilka osób z dawnej redakcji ZINE'a. Drugie i trzecie miejsce jest przyznane niemalże ex aequo. Mnie osobiście zdecydowanie bardziej podoba się STOLEN DATA (ma najlepszy design ze wszystkich magazynów dyskowych na świecie), ale za to McDisk wydawany jest częściej (STOLEN DATA to kwartalnik). Sądzę jednak, że jakość [redacted] końcu zwycięży i STOLEN DATA będzie stanowiło ogromną konkurencję dla R.A.W.-a (który dzięki najciekawszym artykułom ciągle pozostaje moim ulubionym magiem).

1. Sanity - „Jesterday"	230	18,65 %	(1)
2. Anarchy - „Legalise It"	124	10,06 %	(2)
3. Noiseless [redacted] - „Dizzy Tunes"	98	7,95 %	(6)
4. Sanity - „Turmoil"	85	6,89 %	(4)
5. Anarchy - „Legalise [redacted] II"	82	6,65 %	(NEW)
6. Anarchy - „Spring Melodies"	79	6,41 %	(7)
7. Phenomena - „Crystal Symphonies II"	76	6,16 %	(3)
8. Phenomena - „Crystal Symphonies"	52	4,22 %	(5)
9. Parasite - „Imperial Tunes"	46	3,73 %	(16)
Anarchy - „Bruno's Music [redacted] 3"	46	3,73 %	(9)
11. Dual Crew [redacted] Complex - „Finlandia"	36	2,92 %	(12)
12. Phenomena - „Music Dream II"	29	2,35 %	(13)
13. Paradise - „Techno Tower"	21	1,70 %	(8)
14. Hardline - „Black Energy"	19	1,54 %	(15)
15. Silents - „Sound [redacted] Silents"	15	1,22 %	(17)
16. Crusaders - „Bass-O-Matic"	14	1,14 %	(11)
17. Vanish - „Terminate You"	12	0,97 %	(NEW)
Freestyle - „Perfect 5"	12	0,97 %	(10)
19. Kefrens - „Multi Megamix II"	10	0,81 %	(NEW)
[redacted] Chrome - „Digital Disco II"	[redacted]	0,65 %	(18)
Mahoney & [redacted] - „His [redacted] Noise"	8	0,65 %	(13)
Phenomena - „Music Dream"	8	0,65 %	(NEW)
[redacted] Productions - „Musical Secrets"	8	0,65 %	(18)

W tej dziedzinie na topie jest SANITY dzięki swojej nowej idei music-dema. To połączenie świetnej demkowej czołówki i świetnej muzyki dało naprawdę rewelacyjny efekt (podobnie [redacted] przypadku „Turmoilu"). Na dalszych miejscach dyski muzyczne z modułami NUKE'a („Legalise [redacted] I,II") i osławione (jeszcze niedawno [redacted] topie) „Crystal Symphonies I,II".

NAJLEPSZE ART-DYSKI

1. [redacted] Design - „Prism"	252	32,18 %	(5)
2. Kefrens - „Masterpieces"	151	19,28 %	(2)
3. Mirage - „Forgotten"	103	13,15 %	(3)
4. Alcatraz - „Museum"	96	12,26 %	(1)
5. X-Trade - „Color Crime"	78	9,96 %	(4)

Niestety tego typu produktów nie ma na razie zbyt wiele. No cóż, [redacted] końcu nie każdy grafik jest [redacted] stanie wystawić dwadzieścia rewelacyjnych prac, tak jak to zrobił MACK (autor grafiki między innymi [redacted] „Substance" QUARTEXu) [redacted] „Prism" czy R.W.O. w „Masterpieces". Może w niedalekiej przyszłości i nasi graficy popiszą się [redacted] tego typu produktach?

I to tyle na dzisiaj, w przyszłym miesiącu zapraszam [redacted] skosztowanie rarytasów Eurochartsów - notowania najlepszych koderów, grafików, muzyków i... coś jeszcze.

NINJA/UNION

ASEMBLER 68000 (cz. 6)

Ekran Amigi - bojów ciąg dalszy

Witam Was po [illegible] kolejny [illegible] naszym cyklu. Dzisiaj kilka słów [illegible] przewijaniu.

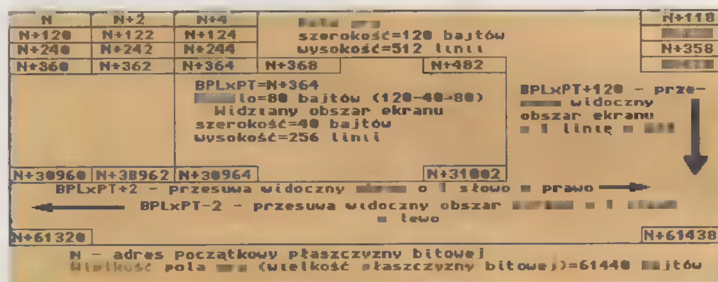
Na początek [illegible] przewijaniu pionowym. Zaraz, a co to jest przewijanie - może ktoś zapytać. Zamiast „przewijanie" możemy powiedzieć „przesuwanie", a tego już wyjaśniać chyba nie trzeba. Na temat przesuwania ekranu [illegible] pionie nie chcę się specjalnie rozwodzić - jeżeli ktoś dokładnie prześledził zamieszczony [illegible] poprzednim odcinku przykładowy program („C&A" 1/93), będzie wiedział [illegible] [illegible] chodzi. Dla przypomnienia podam, że wystarczy zmieniać adresy płaszczyzn bitowych [illegible] wartość równą szerokości płaszczyzny (ilość bajtów [illegible] linii). Adresy płaszczyzn mogą być zmieniane jedynie poza obszarem widzialnym (gdy elektron osiągnie koniec ramki [illegible] prawym dolnym rogu lub na końcu każdej linii ekranu podczas przeskoku do niżej leżącej linii), dlatego też najlepiej nadaje się do tego Copper-lista. Wartości [illegible] programie Coppera muszą być zmieniane [illegible] odpowiednim czasie a nie [illegible] momencie pobierania przez Coppera komendy.

Jak przesuwać ekran [illegible] pionie już wiemy, teraz kolej na przewijanie poziome. Wcześniej musimy zapoznać się z trzema dodatkowymi rejestrami:

BPL1MOD $108 (tylko do zapisu),
BPL2MOD $10A (tylko do zapisu),
BPLCON1 $102 (tylko do zapisu).

Pierwsze dwa rejestry określają wartość tzw. „modulo"; pierwszy (BPL1MOD) dla nieparzystych płaszczyzn bitowych (bitplane: 1, 3, 5), drugi dla płaszczyzn parzystych (bitplane: 2, 4, 6).

Co to jest „modulo"? W celu wyjaśnienia tego pojęcia posłużę się przykładem. Wyobraźmy sobie, że napisaliśmy grę. Pole, na którym ona się rozgrywa, nie mieści się [illegible] całości na ekranie (jest od niego np. wielokrotnie szersze) - nasze pole ma 120 bajtów szerokości [illegible] na ekranie możemy wyświetlić jedynie 40 bajtów [illegible] linii. Amiga wyświetlając linię [illegible] ekranie zwiększa kolejno adresy [illegible] odpowiednich rejestrach BPLxPT (adres wskazujący na słowo płaszczyzny bitowej, z którego aktualnie tworzony jest na ekranie obraz), tak więc po dojściu do końca linii adres jest większy [illegible] 40 bajtów od adresu początkowego. Z prostego rachunku wynika (120-40=80), że aby teraz [illegible] ekranie pojawiła się następna linia naszego pola gry, do adresu płaszczyzny musi zostać dodana wartość 80. I właśnie „modulo" jest taką wartością pozwalającą nadać obszarom pamięci charakterystykę prostokątną określając, jaka liczba ma zostać dodana do adresu płaszczyzny, aby następna linia ekranu była odpowiednią linią pola gry. Pozwala to, wewnątrz dużego obszaru pamięci podzielonego na rzędy (linie) i kolumny, zdefiniowac mniejszy obszar [illegible] odmiennej wysokości i szerokości. Opisany przykład ilustruje rysunek.



Mamy szerokie pole gry (120 bajtów szerokości) [illegible] wysokości np. 512 linii. Szerokość okna wizyjnego wynosi 40 bajtów, wysokość 256 linii. Wartość „modulo": 120-40=80 bajtów (zawsze musi być liczbą parzystą). Jeżeli teraz chcemy wyświetlić [illegible] ekranie lewą górną część naszego pola gry, wystarczy, że jako adres podamy adres początkowy pierwszej linii pola gry. Teraz jeżeli adres [illegible] BPLxPT będziemy zwiększać [illegible] 2 bajty, [illegible] okno wizyjne będzie przesuwać się w prawo nad polem gry. Możemy przesunąć się [illegible] prawo maksymalnie [illegible] 80 bajtów.

Wpiszcie zamieszczony poniżej program, ktory przedstawia omówioną przed chwilą metodę (przesuwa pole [illegible] dwa bajty).

```
START:
    ; ta część programu zarezerwowany obszar pamięci
    ; wypełnia szachownicą
    lea       BITPLANE(pc),a0
    move.l    a0,d0
    lea       ADRBITPLANE(pc),a0
    move.l    d0,(a0)
    lea       BITPLANE(pc),a0
    move.w    #16-1,d7
KWAp2:
    move.w #30*16-1,d6
KWAp1:
    move.w #$ffff,(a0)+
    adda.l    #2,a0
    dbf       d6,KWAp1
    adda.l    #120*16,a0
    dbf       d7,KWAp2
    lea       BITPLANE(pc),a0
    adda.l    #120*16+2,a0
    move.w    #16-1,d7
KWAp4:
    move.w #30*16-1,d6
KWAp3:
    move.w #$ffff,(a0)+
    adda.l    #2,a0
    dbf       d6,KWAp3
    adda.l    #120*16,a0
    dbf       d7,KWAp4

   lea      COPPERLIST(pc),a0 ; adres Copper-listy
   move.w   #$01ff,$dff096      ; zamknięcie kanałów DMA
   move.l   a0,$dff080          ; ustawienie adresu Copper-listy
   clr.w    $dff088             ; wyzerowanie rejestru COPJMP1

   move.w   #$2981,$dff08e      ; wartość dla DIWSTRT
   move.w   #$29c1,$dff090      ; wartośc dla DIWSTOP
   move.w   #$d0,$dff094        ; wartość dla DDFSTOP

   move.w   #$aaa,$dff180       ; kolor tła
   move.w   #$00f,$dff182       ; kolor punktów

   move.w   #$1200,$dff100      ; BPLCON0 - LORES,COLOR,
                                ; 1 bitplane
   move.w   #0,$dff102          ; BPLCON1
   move.w   #0,$dff104          ; BPLCON2

POCZATEK:   ;****************************************
   move.w   #$38,$dff092        ; wartość dla DDFSTRT
   move.w   #80,$dff108         ; BPL1MOD - wartośc modulo
   move.w   #80,$dff10a         ; BPL2MOD /
   move.w   #$8380,$dff096      ; włączenie kanału DMA dla
                                ; Coppera i bitplanów
MOUSE:                        ; przesuw w prawo okna wizyjnego
    bsr       WPISADRES       ; wpisanie adresu do Copper-listy
    move.l    ADRBITPLANE,d0
```

```
    cmp.l    #BITPLANE+120-40,d0 ; czy koniec pola gry
    beq.w    MOUSEp1         ; jeśli tak to przesuw w lewo
    lea      ADRBITPLANE(pc),a0
    add.l    #2,(a0)
    bsr      wolniej
    btst     #$6,$bfe001       ; oczekiwanie na LMB
    bne.b    MOUSE             ; jeżeli nie wciśnięty to skok
                              ; do etykiety MOUSE
    bra.w    WYJSCIE
MOUSEp1:                     ; przesuw w lewo okna wizyjnego
    bsr      WPISADRES
    move.l   ADRBITPLANE,d0
    cmp.l    #BITPLANE,d0  ; czy początek pola gry
    beq.w    MOUSE       ; jeśli tak to przesuw w prawo
    lea      ADRBITPLANE(pc),a0
    sub.l    #2,(a0)
    bsr      wolniej
    btst     #$6,$bfe001       ; oczekiwanie na LMB
    bne.b    MOUSEp1             ; jeżeli nie wciśnięty to skok
                              ; do etykiety MOUSE
    bra.w    WYJSCIE

wolniej:move.w    #6,d6
wolp1:    bsr     wait255
    dbf     d6,wolp1
    rts
KONIEC:   ;*************************************

WYJSCIE:
    move.l   $4,a6               ; EXECBASE do A6
    lea      GRlib(pc),a1        ; nazwa biblioteki w A1
    jsr      -408(a6)            ; otworzenie biblioteki
    move.l   d0,a0
    move.w   #$01ff,$dff096      ; wyłączenie kanałów DMA
    move.l   $26(a0),$dff080     ; wpisanie adresu systemowej
                                 ; Copper-listy
    clr.w    $dff088
    move.w   #$83ff,$dff096      ; włączenie wszystkich kanałów
                                 ; DMA
    move.l   d0,a1               ; bazowy adres biblioteki w A1
    jsr      -414(a6)            ; zamknięcie biblioteki
    rts                          ; wyjscie z programu

WPISADRES:                       ; wpisanie adresu okna wizyjnego
    bsr      WAIT255             ; do Copper-listy
    move.l   ADRBITPLANE,d0
    lea      COPPERLIST(pc),a0
    move.w   d0,6(a0)
    swap     d0
    move.w   d0,2(a0)
    rts
WAIT255:         ; oczekiwanie na koniec ramki
        cmpi.b #$ff,$dff006
        bne.w  WAIT255
WAITp1: cmpi.b #$30,$dff006
        bne.w  WAITp1
        rts
GRlib:  dc.b    'graphics.library',0 ; nazwa biblioteki
     even                          ; instrukcja ASM-ONEa
                                   ; (parzysty adres)
COPPERLIST:              ; początek Copper-listy
    dc.w     $0e0,$0    ;adres płaszczyzny 1 (bity 16-18)
    dc.w     $0e2,$0    ;adres płaszczyzny 1 (bity 0-15)
    dc.w     $ffff,$fffe; instrukcja nielegalna
                        ; (koniec Copper-listy)
ADRBITPLANE:     dc.l    0    ; adres słowa w lewym górnym
                              ; rogu okna wizyjnego
BITPLANE:
    blk.b    120*512,0   ; zarezerwowany obszar pod pole gry
```

Efekt uzyskany na ekranie nie jest na pewno zadowalający: pole przesuwa się skokowo, w krokach co 16 pikseli. Jak więc spowodować, aby pole przesuwało się płynnie, w krokach jednopikselowych? Wspomniałem na początku o trzech rejestrach, poznaliśmy na razie tylko dwa z nich (BPL1MOD i BPL2MOD). Teraz zajmiemy się trzecim rejestrem BPLCON1 ($dff102 - tylko do zapisu).

Jak się zapewne domyślacie, służy on do płynnego przewijania ekranu. Używanych jest tylko 8 niższych bitów tego rejestru (bity 0-7). Za pomocą bitów 0-3 określamy przesunięcie parzystych płaszczyzn bitowych, a bitami 4-7 - płaszczyzn nieparzystych. Przesunięcie może przejmować wartości od 0 do 15 (zapisane jest na 4 bitach dla określonych płaszczyzn bitowych). Wartość ta opóźnia wyjście danych odczytanych z płaszczyzn bitowych.

Płynne przewijanie pola gry w prawo (a okna wizyjnego nad tym polem w lewo), realizuje się zwiekszając wartości w BPLCON1 dla odpowiednich płaszczyzn od 0 do 15. Po osiągnięciu wartości 15 ponownie ustawiamy ją na 0 z jednoczesnym pomniejszeniem BPLxPT o jedno słowo (odjęcie od adresu płaszczyzny bitowej 2 bajtów).

Przesuwanie w lewo uzyskuje się odwracając te czynności. Wartości w BPLCON1 zmniejszamy od 15 do 0. Po osiągnięciu wartości 0 zmieniamy ją ponownie na 15 z jednoczesnym zwiększeniem BPLxPT o jedno słowo (dodając do adresu płaszczyzny bitowej 2 bajty).

Zmieniając wartość w BPLCON1 należy pamiętać, że można tego dokonać jedynie poza obszarem widzialnym (podczas przerwania wygaszania pionowego lub za pomocą Copper-listy). Wartość w BPLCON1 powoduje opóźnienie w wysyłaniu danych każdej linii na ekran, dlatego też aby piksele pojawiające się z lewej strony były widoczne (a nie obcinane), należy zmienić wartość w DDFSTRT. Wartość w tym rejetrze musi zostać ustawiona na wcześniejszą: w niskiej rozdzielczości poziomej (320 punktów w linii) musimy odjąć 8 od normalnie obliczonej wartości dla tego rejestru, w wysokiej rozdzielczości (640 punktów w linii) odejmujemy 4. Musi zostać wtedy także zmieniona wartość „modulo" (od normalnie obliczonej wartości odejmujemy 2) ponieważ odejmując od DDFSTRT zwiększamy szerokość okna wizyjnego. Jeżeli normalnie w linii było wyświetlanych 20 słów, teraz po odjęciu od DDFSTRT wartości 8, dla linii będzie odczytywanych 21 słów.

Znajdujące się poniżej linie należy dopisać do poprzedniego programu zamiast linii znajdujących się pomiędzy etykietami: POCZATEK i KONIEC.

```
POCZATEK:     ;***************************************
    move.w   #$38-8,$dff092     ; wartość dla DDFSTRT
    move.w   #80-2,$dff108      ; BPL1MOD  -wartość modulo
    move.w   #80-2,$dff10a      ; BPL2MOD /
    move.w   #$8380,$dff096     ; włączenie kanału DMA dla
                                ; Coppera i bitplanów
MOUSE:   move.w #$00ff,d7
MOUSEp2: bsr      WPISADRES
    move.w   d7,$dff102      ;przesunięcie do BPLCON1
    tst.w    d7
    beq.w    MOUSEp1
    sub.w    #$0011,d7        ; zmniejszenie przesunięcia
    bra.w    MOUSEp2

MOUSEp1: move.l ADRBITPLANE,d0
    cmp.l    #BITPLANE+120-40,d0
    beq.w    MOUSEl1
    lea      ADRBITPLANE(pc),a0
    add.l    #2,(a0)
    btst     #$6,$bfe001       ; oczekiwanie na LMB
    bne.b    MOUSE             ; jeżeli nie wciśnięty to skok
                               ; do etykiety MOUSE
    bra.w    WYJSCIE
MOUSEl1: clr.l d7
MOUSEl3: bsr     WPISADRES
    move.w   d7,$dff102      ;przesunięcie do BPLCON1
    cmp.w    #$00ff,d7
    beq.w    MOUSEl2
    add.w    #$0011,d7       ;zwiększenie przesunięcia
    bra.w    MOUSEl3

MOUSEl2: move.l ADRBITPLANE,d0
    cmp.l    #BITPLANE,d0
    beq.w    MOUSE
    lea      ADRBITPLANE(pc),a0
    sub.l    #2,(a0)
    btst     #$6,$bfe001     ; oczekiwanie na LMB
    bne.b    MOUSEl1         ; jeżeli nie wciśnięty to skok
                             ; do etykiety MOUSE
KONIEC:     ;************************************
```

(cdn.)

BARTOSZ SMAGA „SMUGGLER"

Jak Wywoływać Duchy?



(odsłona pierwsza)

Każdy, kto choć raz widział działający program na Amidze, widział prawdziwego ducha! No, może nie zupełnie prawdziwego, ale przynajmniej udającego takiego. Kiedy to widzieliśmy? Choćby podczas poruszania strzałki z Workbencha za pomocą myszy lub w dowolnej grze zręcznościowej. Właśnie strzałki, rakiety, ludziki, piłeczki i inne obiekty lubiące się poruszać są najczęściej tym, co wielu przyprawia o gęsią skórkę, czyli duchami. Właściwie powinno się mówić krasnoludki a nie duchy, bo oryginalna nazwa tych obiektów brzmi „sprite" (czyli takie małe z wielką czapą, które macie pod szafą, a nie jest to bałagan). Osobiście uważam jednak, że nazwa „duszek" bardziej pasuje do tych obiektów, albowiem jest zgodna z ich naturą. Duchy są widoczne, a mimo to nie zmieniają zawartości zajmowanej przestrzeni, mogą latać, przenikać przez ściany, znikać i pojawiać się w dowolnym miejscu. Komputerowe duchy czyli sprajty również mają takie możliwości, mówiąc krótko są to małe elementy graficzne, które można używać niezależnie od płaszczyzn bitowych (ang. bitplane). Przystąpmy więc do oswojenia takiego ducha. Aby można było go zobaczyć, musi on mieć

Kolory

Wybór kolorów dla sprajtów jest bardzo podobny do wyboru kolorów dla ekranu w trybie „dual-playfield". W miejscu, w którym wszystkie bity danych sprajta są równe zeru, obiekt jest przezroczysty. Każdy sprajt składa się z dwóch map bitowych, których zawartość określa numer koloru. Sprajty mogą być wyświetlane w dwóch trybach (cztery lub szesnaście kolorów), z których jeden (najniższy) jest przezroczysty. W pierwszym trybie rejestry kolorów są przydzielone dla par sprajtów. Oznacza to, że takie same kolory będzie miał obiekt o numerze 0 i 1. Sprawę wyjaśnia tabela 1.

TABELA 1

Duszek nr	Dane	Rejestr koloru
0 i 1	00	przezroczysty
	01	COLOR17
	10	COLOR18
	11	COLOR19
2 i 3	00	przezroczysty
	01	COLOR21
	10	COLOR22
	11	COLOR23
4 i 5	00	przezroczysty
	01	COLOR25
	10	COLOR26
	11	COLOR27
6 i 7	00	przezroczysty
	01	COLOR29
	10	COLOR30
	11	COLOR31

W trybie szesnastu kolorów wszystkie sprajty mają te same kolory. Jednak aby je uzyskać, musimy połączyć je parami parzysty-nieparzysty (trzeba uzyskać czterobitową mapę kolorów). Kolory dla sprajtów zajmują rejestry od COLOR16 do COLOR31. W tym przypadku kolor 16 jest przezroczysty, a wszystkie szesnastokolorowe duszki mają tę samą paletę kolorów.

[illegible] danych

Jak już pewnie zdołaliscie się zorientować, sprajtów jest osiem. Ich dane są wysyłane poprzez kanały DMA jednak nie można włączyć tylko kilku z nich - trzeba od razu całą ósemkę. Szerokość każdego duszka jest ograniczona do 16 pikseli w niskiej rozdzielczości, ponieważ tyle bitów zawiera jedno słowo. Wysokość ograniczona jest jedynie rozmiarem okna ekranu.

Konstrukcja danych sprajtów jest następująca: dwa słowa kontrolne, które zawierają horyzontalną pozycję sprajta i jego pierwszą oraz ostatnią linię, dane i następne słowa kontrolne itd. Jeśli oba zawierają zera, wówczas DMA kończy operację. Wynika z tego, że jeśli nie chcemy używać wszystkich sprajtów, to nieużywane muszą mieć zera jako pierwsze dwa słowa kontrolne. DMA nie wyśle wtedy żadnych danych. A oto jak wyglądają formaty słów kontrolnych:

pierwsze słowo:

bit nr	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
funkcja	E7	E6	E5	E4	E3	E2	E1	E0	H8	H7	H6	H5	H4	H3	H2	H1

drugie słowo:

bit nr	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
funkcja	L7	L6	L5	L4	L3	L2	L1	L0	AT	0	0	0	0	E8	L8	H0

H0 do H8 - horyzontalna pozycja duszka (HSTART),
E0 do E8 - pierwsza linia duszka (VSTART),
L0 do L8 - ostatnia linia duszka + 1 (VSTOP),
AT - bit przyłączenia.

Poniższy przykład przedstawia dane dla sprajta o wysokości siedmiu linii, którego lewy górny wierzchołek leży na pozycji 180,160.

Duszek:

```
dc.w  $A05A,$A700          ;HSTART=$B4, VSTART=$A0, VSTOP=$A7
;          bity 0                bity 1 koloru
dc.w  %1111111111111111,%1111111111111111
dc.w  %1000000000000001,%1111111111111111
dc.w  %1011111111111101,%1100000000000011
dc.w  %1010000000000101,%1100000000000011
dc.w  %1011111111111101,%1100000000000011
dc.w  %1000000000000001,%1111111111111111
dc.w  %1111111111111111,%1111111111111111
dc.w  0,0          ;koniec danych dla sprajta
```

Po nałożeniu słów danych duszka otrzymamy prostokątną ramkę, w której kolory są rozmieszczone następująco:

```
3333333333333333
3222222222222223
3211111111111123
3210000000000123
3211111111111123
3222222222222223
3333333333333333
```

Kolor 0 jest oczywiście przezroczysty. Zauważyliście pewnie, że w drugim słowie kontrolnym znajduje się bit przyłączania AT. Służy on do łączenia ze sobą dwóch duszków. Po co? Dzięki temu uzyskamy czterobitową mapę kolorów. Po co? Na czterech bitach można zapisać liczby od 0 do 15. Już coś świta? Oczywiście! W ten sposób uzyskamy szesnastokolorowego duszka. Takich duszków można uzyskać maksymalnie cztery, ale nie można przecież mieć wszystkiego naraz.

Tu kilka ważnych uwag. Bit AT działa tylko dla sprajtów nieparzystych. Po jego ustawieniu otrzymujemy pary duszków 0-1, 2-3, 4-5 i 6-7. Pozycje par sprajtów MUSZĄ BYĆ TAKIE SAME. Jeśli będą się różniły choć o maciupeńki piksel, to duszek zostanie wyświetlony w trybie czterokolorowym. Poniżej przedstawiam przykład danych duszka na pozycji 180,160 w szesnastu kolorach. Będzie się on składał z jednej linii ale za to bardzo kolorowej (o kolejnych kolorach od 0 do 16).

Duszek0:

```
dc.w  $A05A,$A100          ;HSTART=$B4, VSTART=$A0, VSTOP=$A1
;      bity zerowe koloru     bity pierwsze koloru
dc.w  %0101010101010101,%0011001100110011
dc.w  0,0
Duszek1:
```

```
dc.[illegible]  $A05A,$A180          ;HSTART=$B4, VSTART=$A0, VSTOP=$A1
;              bity 2 koloru            bity 3 koloru
dc.w    %0000111100001111,%0000000011111111
dc.w    0,0
```

Aktywacja [illegible]

Po skonstruowaniu [illegible] CHIP-RAM poprawnej listy danych duszka i wpisaniu do tabeli kolorów potrzebnych wartości, sterownik DMA musi zostać powiadomiony o adresie listy (przed włączeniem DMA duszków). Każdy kanał DMA duszków [illegible] parę rejestrów, do której musi zostać wpisany adres listy - patrz tabela 2

[illegible] 2

Sterownik DMA używa owych rejestrów jako wskaźników aktualnych

SPRxPT (SPRite [illegible] PoinTer)

Rejestr	Nazwa	Funkcja
$120	SPR0PTH	wskaźnik listy danych duszka, bity 16-18
$122	SPR0PTL	dla kanału DMA 0, bity 0-15
$124	SPR1PTH	wskaźnik listy
$126	SPR1PTL	dla kanału 1
$128	SPR2PTH	wskaźnik listy
$12a	SPR2PTL	dla kanału 2
$12c	SPR3PTH	wskaznik listy
$12e	SPR3PTL	dla kanału [illegible]
$130	SPR4PTH	wskaznik listy
$132	SPR4PTL	dla kanału 4
$134	SPR5PTH	wskaźnik listy
$136	SPR5PTL	dla kanału [illegible]
$138	SPR6PTH	wskaźnik listy
$13a	SPR6PTL	dla kanału 6
$13c	SPR7PTH	wskaznik listy
$13e	SPR7PTL	dla kanału 7

SPR0PTH i SPR0PTL oznaczają odpowiednio starsze i młodsze słowo adresu danych duszka [illegible]

adresów w listach. Na początku każdej klatki zawierają one adres pierwszego słowa sterowania. Po odczytaniu każdego słowa danych są one zwiększane [illegible] jedno słowo, a więc przy końcu obrazu wskazują pierwsze słowo po liście. Wskaźniki te muszą być ponownie ustawiane na początek listy przed każdą klatką. Tak jak [illegible] przypadku wskaźników bitplane'ów BPLxPT, jest to wykonywane przez Coppera [illegible] czasie przerwania synchronizacji pionowej. Odpowiedzialna za to część Copper-listy wygląda tak:

```
DuszekxH = adres początkowy listy danych dla duszka x, bity 16-18
DuszekxL = bity 0-15
CopperlistStart
MOVE #Duszek0H, SPR0PTH  ;włączenie zerowego kanału DMA
MOVE #Duszek0L, SPT0PTL  ;duszka
MOVE #Duszek1H, SPR1PTH  ;włączenie pierwszego kanału DMA
MOVE #Duszek1L, SPT1PTL  ;duszka
MOVE #Duszek2H, SPR2PTH  ,włączenie drugiego kanału DMA
MOVE #Duszek2L, SPT2PTL  ;duszka
 .. ... ...              ;to [illegible] dla kanałów [illegible] do [illegible]
MOVE #Duszek7H, SPR7PTH  ;włączenie siódmego kanału DMA
MOVE #Duszek7L, SPT7PTL  ;duszka
 .. .. ...               ;inne rozkazy
WAIT $fffe               ;koniec Copper-listy
```

UWAGA! Użyte rozkazy MOVE i WAIT nie są instrukcjami asemblera lecz Coppera i należy je zamienić na dane liczbowe.

Po zbudowaniu Copper-listy możemy włączyć DMA duszków [illegible] pomocą rozkazu

```
MOVE.W   #$82A0,$DFF096
```

Ustawia [illegible] bity SPREN, COPEN i DMAEN w rejestrze DMACON.

Myślę, że zamieszczone tu informacje powinny doprowadzić Was do optymistycznego okrzyku: „Łaaa! Duuuch", gdy uzyskacie go na swoim ekranie. Lecz czy można przestraszyć się ducha, który nie potrafi się ruszać? Oczywiście nie. Zatem ducha należy poruszyć. Zrobimy to [illegible] następnym odcinku.

(cdn.)

PRZEMYSŁAW CIEŚLAK

MONEY FOR NOTHING

Czyli jak nie zostać milionerem

Kiedy zaczyna Ci brakować nowych sampli do ulubionego ProTrackera, a tworzone „kawałki" brzmią prawie tak samo to znak, że nadszedł czas na... MIDI. Jego niesamowita elastyczność i uniwersalność umożliwią Ci odkrycie nowych, niezbadanych rejonów muzyki. Może zabrzmiało to patetycznie, ale cyfrowy interfejs instrumentów muzycznych zadomowił się na stałe w pejzażu muzycznym, nawet w tym najbardziej akustycznym. Amiga lub C-64 rozszerzone o przystawki MIDI wraz z odpowiednim oprogramowaniem mogą stać się centrum dowodzenia w Twoim studio muzycznym. Oczywiście trzeba jeszcze dokupić odpowiedni instrument, którego wybór nie jest już taki prosty.

Dla przypomnienia

Interfejs MIDI służy do wymiany danych czyli transmisji KOMUNIKATÓW MIDI między różnego rodzaju urządzeniami np. klawiaturą sterującą (Master Keyboard) ROLAND A-30 i modułem brzmieniowym (Sound Modul) KORG KR 03R/W. Komunikaty MIDI to po prostu ciąg odpowiednio spreparowanych bajtów opisujących parametry odtwarzanego dźwięku czyli wysokość, dynamikę, barwę dźwięku (pianino, skrzypce, kontrabas), dodatkowe efekty (Chorus, Delay), ustawienie w panoramie stereofonicznej itd. Należy uświadomić sobie, że złączem MIDI przesyłane są JEDYNIE PARAMETRY (bity, bajty) powstającej muzyki, zaś generacją dźwięku zajmują się instrumenty (moduły brzmieniowe).

Powstało sporo programów do składowania komunikatów MIDI, a nawet do ich wysyłania - sekwencer. Programy sekwencerowe (OCTAMED, PRO 24 III, PIPES & BARS) można porównać do magnetofonów wielośladowych, gdzie na każdej ścieżce można zapisać dowolną linię melodyczną. Dzięki pracy w czasie rzeczywistym istnieje możliwość dogrywania na żywo nowych parti muzycznych, kasowania, transpozycji o np. oktawę, dzielenia na mniejsze części, zmiana barwy, głośności itd. Inną ważną rzeczą jest dostęp do większości funkcji instrumentu poprzez MIDI (ale o tym innym razem).

Dane przesyłane są w postaci szeregowej od gniazda MIDI OUT do MIDI IN. MIDI THRU retransmituje komunikaty ze złącza MIDI IN do większej liczby urządzeń. Za pomocą jednej klawiatury można na kilku instrumentach zagrać akord E-dur, ale różnymi barwami. Dla rozróżnienia kilku urządzeń pracujących w systemie MIDI wprowadzono pojęcie kanałów - jest ich 16. Ułatwiają one adresowanie danych do poszczególnych instrumentów obecnych w systemie MIDI. Szybkość transmisji danych wynosi 31250 bitów na sekundę i dlatego jednoczesne naciśnięcie kilku klawiszy nie powoduje słyszalnego opóźnienia w generacji dźwięków.

Ci tak [illegible]

Jak wszystkim wiadomo, instrumenty muzyczne nie należą do najtańszych a zakup na potrzeby profesjonalne wiąże się z wydatkowaniem ogromnych kwot pieniędzy, np. YAMAHA TG-500 kosztuje ponad 30 mln zł.

Jednak w klasie instrumentów popularnych można także znaleźć całkiem przyzwoite urządzenia. Dla tych, którzy chcą się tylko pobawić, polecam tanie Home Keyboard firmy YAMAHA, serie PSS, PSR, KAWAI seria MS, ROLAND seria E, CASIO seria CT. Posiadają one z reguły kilkanaście barw o średniej jakości, kilka rodzajów rytmów oraz tradycyjnie MIDI. Jednak możliwości wykorzystania tego ostatniego są raczej niewielkie, np. obsługa 2 lub 4 kanałów, stałe włączenie LOCAL na ON.



„Normalna" wersja syntezatora Roland D-10

Bardziej ambitnym (ale z mniej wypchaną kieszenią) polecam zakup starszych modeli, z reguły o wiele tańszych od nowinek technicznych. Dla przykładu ROLAND D-5, D-10, U-20, YAMAHA SY-22, SY-55, KAWAI K1, CASIO VZ-1. Są to instrumenty o przyzwoitych parametrach użytkowych a wysoki poziom technologiczny wróży im bezproblemową eksploatację jeszcze przez następne 20 lat.

Producenci świadomi, że muzyk nie musi być od razu miliarderem, wprowadzili na rynek okrojone wersje swoich wyrobów, tzw. moduły. Są to po prostu syntezatory, perkusje itp., ale bez urządzeń wejścia czyli klawiatur, padów, wyzwalaczy itp. Cena wersji modułowej typowego syntezatora jest przynajmniej o 20% niższa od normalnej. Ostatnio coraz więcej pojawia się na rynku samych modułów, a dopiero później dobudowuje się do nich klawiatury. Przykład: popularny ROLAND SoundCanvas SC-55 kosztuje około 9 mln zł. Za wersję z klawiaturą tzn. ROLAND JV-30 trzeba zapłacić przynajmniej 16-17 milionów. Rachunek jest prosty.

Dla tych, którzy już posiadają instrument, zakup nowego w postaci modułu jest o wiele ekonomiczniejszy (niższa cena) i wygodniejszy (mniej miejsca). Brak klawiatury nie oznacza, że wytwórca obniżył parametry użytkowe. Czasem wręcz przeciwnie. YAMAHA SY-22 ma 16-głosową polifonię. Jej odpowiednik modułowy YAMAHA TG-33 ma już 32 głosy. Sterowanie modułu odbywa się w głównej części po złączu MIDI poprzez wysłanie do urządzenia komunikatów MIDI np. zagraj akord D-dur, w kanale 1, barwą numer 49 (skrzypce), z maksymalną dynamiką (127) i dodanym efektem Chorus (KONTROLER 93) o 50% głębokości (64). Można również wpływać na barwę tworzonej muzyki przez sterowanie syntezatora panelem kontrolnym. W zależności od modelu może on być bardziej (ROLAND SC-155) lub mniej (YAMAHA TG-100) rozbudowany.

Użytkownicy Amigi i C-64 szczególnie przyzwyczajeni do zapisu trackerowego, a chcący oszczędzić trochę pieniędzy, powinni nabyć instrumenty w wersji modułowej (z ang. RACK-owej), klawiatura muzyczna i tak będzie rzadko używana. W razie przypływu gotówki można rozbudować swoją bazę sprzętową o klawiaturę sterującą np. ROLAND seria PC i A, YAMAHA - KX.

Tradycyjnie przy zakupie instrumentu musimy niestety pójść na pewien kompromis między tym co CHCEMY, a tym co MOŻEMY kupić. Generalnie kierujemy się parametrami użytkowymi danego urządzenia.

[illegible] parametry instrumentu

Na poziomie równorzędnym stoją dwa: POLIFONIA (POLYPHONY) i wielobrzmieniowość (MULTITIMBRAL). Polifonia określa ile głosów równocześnie może wygenerować instrument.Obecnie średnia wynosi 28 (YAMAHA TG-500 64 głosy).

Wielobrzmieniowość określa liczbę barwy, jaka może jednocześnie odezwać się w danej chwili (typowo 16). Jest to szczególnie ważny parametr, gdy instrument będzie służył jako podstawowa baza brzmieniowa. Jeśli kupujemy urządzenie tylko do grania na estradzie (PLAYBACK), MULTITIMBRAL w zasadzie nas nie obchodzi i może być nieco mniejszy od średniej światowej. Za to liczba rodzajów brzmień i polifonia powinny wyrażać się najwyższą liczbą. Znakomity ROLAND D-50 potrafi odezwać się tylko dwoma różnymi barwami ale do dziś jest używany przez wielu muzyków.

Kolejnym ważnym parametrem jest dostępna liczba barw. Dla instrumentów średniej klasy np. ROLAND SC-55 wynosi 314, oprócz [illegible] zestawów perkusyjnych. Równorzędna jest tu jakość generowanych brzmień co wpływa na sposób, a nawet na rodzaj wykonywanej muzyki.

Większość obecnie produkowanych syntezatorów-

torów wyposażono w procesor efektów. Jest to takie urządzenie, które włączone w tor elektroakustyczny powoduje nałożenie charakterystycznych „zniekształceń" np. dodanie pogłosu, chorusa, flangera, przesunięcia fazowego (stosowanego często przez J.M. Jarre'a, płyta OXYGEN), equalizera itp. Tradycyjnie im więcej, tym lepiej.

Jeżeli w perspektywie zamierzamy przenieść naszą muzykę na magnetofon wielośladowy, ważnym staje się liczba niezależnych wyjść w instrumencie. Oznacza to MOŻLIWOŚĆ przyporządkowania różnym wyjściom poszczególnych kanałów MIDI lub nawet barw. Przykładowo na pierwszym wyjściu wysyłamy bas, na drugim perkusję, trzeci obsługuje solo. W wyniku tego na każdą partię nagrywanej muzyki możemy wpływać w nieograniczony sposób dodając w zależności od potrzeb więcej tonów niskich, echa z zewnętrznego procesora efektów.

Dla „majsterkowiczów", którzy lubią eksperymentować z nowymi brzmieniami, instrument powinien zawierać rozbudowany zestaw edycji barw. W sprzęcie popularnym możemy wpłynąć na obwiednię dźwięku (ADSR), ustawienie filtru dolno- i górnoprzepustowego, nałożenie przebiegów wolnozmiennych (tzw. tremolo), pogłos, chorus itd. W bardzo rozbudowanych systemach można mieszać ze sobą próbki (sample playback) z przebiegami z tradycyjnych generatorów.

Warto również zwrócić uwagę na możliwość dodawania nowych brzmień i próbek poprzez podłączenie karty PCM (płaska karta pamięciowa wielkości wizytówki) oraz zapisywania własnych barw również na karcie PCM. Dla własnej wygody i zdrowia trzeba wziąć pod uwagę rodzaj wyświetlacza, jakim instrument posługuje się do komunikacji z użytkownikiem.

Wersja modułowa Rolanda D-10

Najlepsze są typu LCD, podświetlane zestawem pomarańczowych diod LED. Stanowczo odradzam wyświetlacz typu fluorescencyjnego (np. ENSONIQ SD-1), przy którym praca jest dosyć uciążliwa. Podobnie jak obraz w telewizorze wyświetlacz odświeżany jest 50 razy na sekundę, co przy jaskrawej, zielonej barwie wywołuje prawdziwy oczopląs. Dla wersji modułowych instrumentów ważnym parametrem jest możliwość przesyłania przez interfejs MIDI jak największej liczby różnych typów kodów sterujących tzw. KONTROLERÓW. Umożliwiają one zdalne sterowanie urządzeniem, przez MIDI - karta implementacyjna instrumentu.

Moim zdaniem omówiony zestaw parametrów jest absolutnym minimum, o którym przyszły użytkownik syntezatora lub perkusji elektronicznej powinien wiedzieć. Ciężko zarobionych pieniędzy nie powinno się lokować w „profesjonalnym" złomie.

Na zakończenie

Co za dużo to... zdrowo. Nieco przewrotne ujęcie starego porzekadła ludowego odzwierciedla wagę problemu. Im więcej będziesz miał pod ręką urządzeń, a więc brzmień, tym Twoja muzyka będzie pełniejsza i mniej mechaniczna. Ideałem jest przydzielenie do każdej grupy instrumentów oddzielnego modułu np. perkusja - ALESIS D4, 7 mln zł, ponad 500 (!) brzmień perkusyjnych i efektów specjalnych najwyższej jakości, brzmienia „kosmiczne" - E-MU PROTEUS 1XR, 900$, 128 barw, itd.

Do zabawy w sampling (RAP, HOUSE, HIP HOP) jakości studyjnej należy pomyśleć o samplerze np. ROLAND W30 lub... ATARI FALCON 030 (około 26 mln zł). Tradycyjnie limituje nas stan konta, ale dla chcącego nie ma nic trudnego. Należy pamiętać, że najważniejszym elementem systemu MIDI jesteś Ty, Twoja wyobraźnia muzyczna i umiejętności.

WŁADCA NIEPODZIELNY MIDIlandu

Robert MM Chojecki

P.S. Przy nieco poważniejszej „zabawie" w muzykę powstaje problem jakości generowanych brzmień. Dostępne w Twojej Amidze cztery kanały 8-bitowe lub SID w C-64 to trochę za mało jak na warunki studyjne. Amigę można rozszerzyć o dodatkowe przetwornik, ale 16-bitowa karta firmy SunRize kosztuje około 1489 $ czyli tyle, ile dobry, markowy syntezator np. KORG M1R. Chociaż... nasz redakcyjny kompozytor BAD potrafi i na zwykłej A500 komponować extra muzyczki.

Mysz DATALUX SV-713

Gdyby nie ludzka pomysłowość pewnie do dzisiaj mieszkalibyśmy w jaskiniach lub ewentualnie w lepiankach. Stwierdzenie to odnosi się również do komputerów. Postęp technologiczny pozwolił m.in. na to, że w Twojej Amidze zawarta jest moc obliczeniowa praprzodka dzisiejszych komputerów ENIAC-a, który zajmował czteropiętrowy budynek i pobierał moc rzędu megawatów. Oczywiście konstruktorzy miewają czasem i inne ciekawe pomysły. Ot, chociażby dwóch panów z USA wymyśliło sobie urządzenie o nazwie mouse czyli mysz.

Amiga wyposażona jest standardowo w graficzny system porozumiewania się z użytkownikiem. Pierwszoplanową rolę odgrywa tu oczywiście nasza stara, znajoma mysz. Bez niej jakakolwiek praca z komputerem byłaby nie do pomyślenia. Wskaźnik myszy znajdujący się na ekranie zastępuje nasze palce, którymi wciskamy/wskazujemy wymyślne gadżety, guziki, przyciski itp., natomiast ekran spełnia rolę w pełni interaktywnej konsoli, pulpitu sterowniczego.

O ile przez pierwszy okres użytkowania Amigi jesteśmy zafascynowani możliwościami gryzonia standardowo dołączanego do komputera, to jednak po pewnym okresie dochodzimy do wniosku, że mysz ta mogłaby być lepiej zaprojektowana. Jeżeli dodatkowo odczuwamy okresowe bóle prawej ręki (leworęczni lewej) w okolicach łokcia, to należy pomyśleć o zakupie nowego gryzonia.

Myszy ci u nas dostatek, jak powiada wujek Miecio, i dlatego pragnę zaprezentować Wam



produkt firmy DATALUX o symbolu SV-713. Mysz jest trochę nietypowa, ponieważ ma obudowę wykonaną z przezroczystego tworzywa sztucznego (patrz zdjęcie). Można sobie popatrzyć do woli na jej wnętrzności, także w czasie pracy. Kto nie znał do tej pory zasady działania tego urządzenia, z pewnością pozna ją po kilkugodzinnej sesji z myszą. Tak, tak, to właśnie ruch malutkich kółek ze szprychami (które poruszane są przez obracające się wałeczki, a te z kolei wprawia w ruch kula) jest wychwytywany przez elementy fotoelektryczne i zamieniany na impulsy elektryczne. SV-713 pracuje więc na zasadzie optomechanicznej, zatem pod względem mechanicznym jest identyczna jak amigowska. Oczywiście fakt, iż SV-713 ma przezroczystą obudowę dla jednych może być zaletą, dla innych wadą (widok mechanizmu może rozpraszać podczas pracy), na pewno jednak jest to pomysł oryginalny i mnie osobiście bardzo się podoba.

DATALUX MOUSE zdecydowanie lepiej pasuje do ludzkiej dłoni, niż mysz firmy Commodore. Krzywizny są bardziej opływowe, mysz daje się wygodnie uchwycić. Przyciski, w przeciwieństwie do standardowej myszy, mają wyraźny skok. Każdemu naciśnięciu towarzyszy charakterystyczny, cichy trzask, tzw. kliknięcie. Ułatwia to znakomicie pracę. Również przewód łączący urządzenie z Amigą jest bardziej elastyczny od produktu firmy Commodore. Rozdzielczość myszy jest nieco wyższa od standardowej - około 200 dpi.

Ogólnie DATALUX MOUSE SV-713 prezentuje się dobrze, ma dobre parametry i moim zdaniem warta jest zainteresowania użytkowników Amigi. Na zakończenie dodam, że dystrybutor udziela 12 miesięcznej gwarancji.

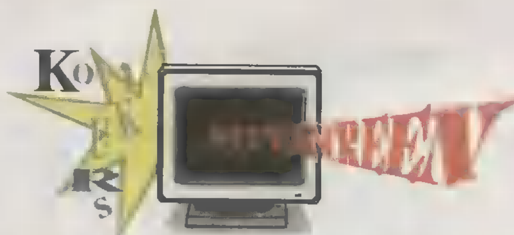
Młodszy Specyalista Do Spraw Łapania Myszy

RMMCH

**DYSTRYBUTOR: „3-STATE POLAND", 04-098 Warszawa,
ul. Majdańska 9, tel./fax 6102892**
CENA: 390.000 zł (może ulec obniżeniu!)

ZALETY:
+ oryginalna obudowa
+ rozdzielczość 200 dpi
+ ergonomiczny kształt
+ długa żywotność

WADY:
- brak informacji po polsku (załączony tylko krótki [illegible]



Czoł(gi)em malarze!



„Kat", [redacted] Supcziński, C-64 (I nagroda)



„Kosmos", Tomasz Szatewicz, Amiga (I nagroda)

Tym [redacted] komodorowcy po prostu poszaleli. Tak, serio. Przysłaliście nam aż 133 obrazki (25 autorów)! Ale [redacted] się dziwić, skoro dyskietki niektórych zawierały po 20 prac. Było też 10 grafik grupy FROGER z Gliwic, niestety niezbyt ciekawych (mam jednak nadzieję, że grafik tej grupy szybko się podszkoli).

Aby nikogo nie skrzywdzić, ocenę części prac przenieśliśmy na następny miesiąc. A [redacted] tej edycji konkur[redacted] nagrody otrzymują (kategoria C-64):

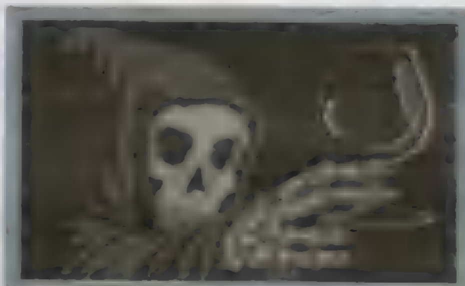
1. **Bartosz Supcziński**, Toruń - I nagroda (MEGAPAKIET produkcji FET) [redacted] prace pt. „Kat" i „Na zdrowie".
2. **Paweł Harasimowicz**, Słupsk - II nagroda (C&S EDYTOR produkcji FET) za pracę pt. „Mig-29UB". (Szanowny Pawle, to już Twoja druga druga nagroda w naszym konkursie - gratulujemy i czekamy [redacted] rysunki przedstawiające nie tylko samoloty.)
3. **Norbert Łukasik**, Lublin - III nagroda (C&S EDYTOR produkcji FET) za pracę pt. „Zamczysko".

Natomiast amiganci zawiedli znowu. Co prawda nadesłaliście 68 prac (15 autorów), ale obrazki są słabe, zupełnie jakby były malowane na C-64, a nie na Amidze. Jedyny wyjątek to prace Tomasza Szatewicza, który oczywiście dostał jednogłośnie pierwszą nagrodę. Jeden z Czytelników dopuścił się niestety kradzieży, bo przysłał grafiki nie swoje, ściągnięte. Nazwiska wspaniałomyślnie nie opublikujemy, niech ów delikwent się zastanowi i uderzy [redacted] skruchę do własnego sumienia. A [redacted] nagrody:

1. **Tomasz Szatewicz**, Mrągowo - I nagroda (gry „Blue Angel" i „Vampire's Empire") za prace pt. „Kosmos", „Odrzutowiec" i „Potwór".
2. **Damian Witkowski**, Połaniec - II nagroda (gra „Western Games") [redacted] pracę pt. „Galera".

Mam jednak dla amigantów pocieszającą wiadomość. Otóż [redacted] dwa dni po tym, jak redakcyjne jury wybrało już najlepsze prace, listonosz przyniósł trzy świeże przesyłki zawierające [redacted] grafiki na konkurs. Panowie! To było to! Nareszcie prawdziwi mistrzowie, Picasso czy Rubens wysiadają. No, ale o tym [redacted] następnym miesiącu, wszyscy zobaczycie te grafiki, bo z pewnością dostaną one nagrody. No to cześć!

Don Pedro Konkursolini



„Na zdrowie", Bartosz Supcziński, C-64 (I nagroda)



„Odrzutowiec", Tomasz Szatewicz, Amiga (I nagroda)



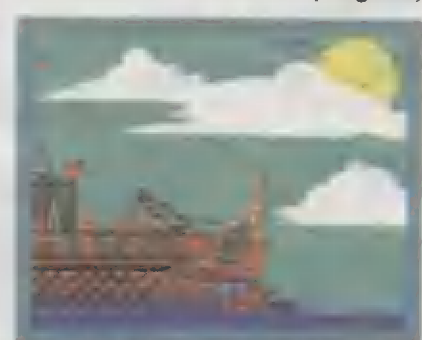
„Mig-29 UB", Paweł Harasimowicz, C-64 (II nagroda)



„Potwór", Tomasz Szatewicz, Amiga (I nagroda)



„Zamczysko", Norbert Łukasik, C-64 (III nagroda)



„Galera", Damian Witkowski, Amiga (II nagroda)



Kupon ważny do 05-07-93

KUPIĘ

Kupię drukarkę igłową, wymienię się oprogramowaniem na PC AT. Artur Olszanski, 20-323 Lublin, ul. Puchacza 9/12, tel. (0-81) 419-42.

Kupię pilnie książkę Wasława Iszkowskiego „Nauka programowania w języku BASIC dla początkujących". Mariusz Posadzy, 88-190 Barcin, ul. Lotników 3/42.

Kupię cartridge do C-16, joystick z okrągłym wejściem oraz instrukcję i wszelką literaturę dotyczącą C-16, oczywiście [redacted] języku polskim. Zbigniew Słowiński, 92-237 Łódź, ul. Kazimierza 11 m 25, telefon 74-78-86.

Kupię: Macroasembler Development System, Merlin 64, Panther, TurboAssembler 5.1, książkę „Jak rozbudować interpreter C-64" Gajewskiego i Radziszewskiego, cartridge Expert, Action Replay 4.0. Tomasz Jastrzębski, 07-409 Ostrołęka, ul. M. Modrzejewskiej 21/49, tel. 65-558.

SPRZEDAM

Sprzedam C-64 (stan idealny), magnetofon, trzy joysticki, moduł Black Box, cartridge [redacted] grami, oprogramowanie na kasetach, literaturę, pokrywę [redacted] klawiaturę. Cena około 1.7 mln zł. Adam Sakowicz, 62-200 Gniezno, ul. Witkowska 68.

Sprzedam C-64 (w dobrym stanie), dwa joysticki, magnetofon, moduł Black Box 8.0, oprogramowanie na kasetach. Cena całości [redacted] mln zł. Mariusz Majewski, 20-857 Lublin, ul. Hamasie 21/66, tel. 71-11-18.

Sprzedam Amigę 500 (1 [redacted] CHIP RAM, 1.8 SLOW RAM, BOOT SELECTOR), modulator, joystick, literaturę. Cena: 6,5 mln zł. Tomasz Dubiecki, 57-401 Nowa Ruda, ul. Świdnicka 92/2.

Sprzedam monitor kolorowy Commodore 1084 S [redacted] literaturę i peryferia [redacted] Amigi. Poza tym oferuję oryginalną [redacted] HARPOON (sygnowaną przez IPS). Paweł Popławski, 96-200 Rawa Mazowiecka, Oś. [redacted] Maja 7/4, tel. 23-72, od godz. 16.

Sprzedam Atari [redacted] XE, magnetofon z systemem TURBO 2000 i 3000 lub zamienię na C-64 u podobnym osprzętem. Dariusz Michalski, 58-506 Jelenia Góra, ul. Kiepury 16 m 15, tel. 42-979

Sprzedam C-64 II, stacja dyskow 1541 II (gwarancja), monitor Philips 12 cali zielony, magnetofon 1535, moduły Final III, Black Box, dwa joysticki, oprogramowanie na kasetach i dyskietkach, pokrywę na klawiaturę, pudełko na dyskietki, filtr do monitora, literaturę. Cena: 4.5 mln zł. Jarosław Zychowicz 57-410 Ścinawka Średnia, Ścinawka Górna 16.

Sprzedam C-64 [redacted] magnetofonem, moduł Black Box 4.0, literaturę. Naprawię uszkodzone zasilacze C-64. Informacja - koperta zwrotna [redacted] znaczkiem. Jacek Gałkowski, 86-105 Świecie, ul. Ks. S. Krausego 17/21.

Sprzedam nowy [redacted] gwarancji C-64, magnetofon, moduł Black Box 4.0, oprogramowanie na kasetach, pokrywę [redacted] klawiaturę, joysticki, literaturę. Cena: oko 2 mln zł. Bartek Jaglarz, 20-337 Lublin, ul. Pogodna 34/36, tel. 483-50.

Sprzedam C-64 II, magnetofon 1530 C2N, dwa joysticki, pokrywę [redacted] klawiaturę, dwa moduły Black Box [redacted] syntezą polskiej mowy [redacted] gry, oprogramowanie [redacted] kasetach, literaturę. Marcin Barańczyk, 67-200 Głogow, ul. Armii krajowej 6/10, tel. (0-70) 33-47-90.

Sprzedam Amigę 500+ (1 MB CHIP RAM, gwarancja), nową mysz, joysticki - 6,5 mln [redacted] 1 MB CHIP RAM do Amigi 500+ (gwarancja) - 900 tys. Janusz Jura, 81-862 Sopot, ul. Mazowiecka 30A/13, tel. 51-92-09.

Sprzedam C-64 (gwarancja), magnetofon 1530 (gwarancja), dwa joysticki, oprogramowanie na kasetach, moduł Black Box 4.0, literaturę. Cena całości 2,3 mln zł. Tomasz Grzegorczuk, Białystok, ul. Bolesła[redacted] Chrobrego 5B/100, tel. 320-973.

Sprzedam C-64, magnetofon, stację dysków [redacted] literaturę, kasety i dyskietki z oprogramowaniem. Cena do uzgodnienia około 3,5 mln zł. Rafał Wroblewski, 02-396 Warszawa, ul. Geodetów 12 m 83, tel. 22-38-11.

Sprzedam C-64, stację dyskow 1541 II, moduł Final III (gwarancja), mysz, pudełko na dyskietki, literaturę. Cena całości 4,9 mln zł. Wszyst[redacted] mogę zamienić na używaną Amigę 500. Dariusz Sieradzki, 10-804 Olsztyn, ul. Rolna 227.

Okazja! Pilnie sprzedam C-64 II, magnetofon, oprogramowanie, pokrywę [redacted] klawiaturę, pudełko [redacted] kasety, cartridge Black Box 4.0, przewod wizyjny DIN - CINCH, telewizor-monitor SANYO. Wszystko [redacted] 4 mln zł. Adam Suchoń, 42-200 Częstochowa, ul. Bohaterów Katynia 46 m 41.

Sprzedam C-128D (metalowa obudowa), monitor NEPTUN 156, moduł Action Plus 6.0, magnetofon, oprogramowanie na dyskietkach, DiskBox, dwa joysticki, literaturę. Cena 6,5 mln zł. Michał Czajkowski, 85-858 Bydgoszcz, ul. Komuny Paryskiej 11/94, tel. 63-85-73.

Sprzedam Commodore C-64 (gwarancja), magnetofon 1530 (gwarancja), moduł X, pokrywę [redacted] klawiaturę, literaturę. Cena: [redacted] mln zł. Artur Białous, 19-230 Szczuczyn, ul. J. Falkowskiego 24, tel. 52-72.

Sprzedam C-64 II (stan idealny), magnetofon, stację dysków 1541 II (gwarancja), moduł Final II, dyskietki, kasety, literaturę. Cena do uzgodnienia. Michał Witkowski, 26-110 Skarżysko Kamienna, ul. Żeromskiego 47/12, tel. 53-77-07.

Sprzedam C-64 (stan idealny), stacja dyskow 9900 (gwarancja), joysticki, moduł Final II, dyskietki [redacted] oprogramowaniem, pudełko na dyskietki, literaturę. Cena 4,2 mln zł. Krzysztof Wardaszko, 05-200 Wołomin, ul. Kobyłkowska 30/6, tel. 76-40-19.Sprzedam kolorowy monitor Commodore 1802 (mało używany), dołączę joystick „Jet Fighter". Cena 3 mln zł. Anna Ryfka, 20-532 Lublin, ul. 20-532 Lublin, tel. 56-43-18.

Sprzedam Amigę 500 1 MB (gwarancja), modulator TV, pokrywę [redacted] komputer, dyskietki, literaturę. Robert Lendzion, 01-874 Warszawa, ul. Duracza 11A/41, po godz. 16.00.

Sprzedam komputer Amiga 500 1 MB RAM, monitor 1048S, słuchawki, joystick, kilkanaście gier, literaturę. Sprzęt na gwarancji do 12.93 roku. Jarosław Skwarski, 80-759 Gdańsk, ul. Mostek 26/23.

Sprzedam C-64, magnetofon, joystick, cartridge Bis Plus, Black Box, oprogramowanie [redacted] kaetach, literaturę, Miniwierzę Audio. Cena 4,5 mln zł. Artur Stolorz, 41-407 Imielin, ul. Niemcewicza 7.

Sprzedam C-64, stacja dysków 1541 II, dyskietki z oprogramowaniem, magnetofon, cartridge Final II, X, moduły z grami, dwa joysticki, literaturę lub całość wymienię [redacted] używaną Amigę 500. Ewentualnie dopłacę. Karol Olicjał, 20-552 Lublin, ul. Kawaleryjska 18/2.

Sprzedam C-64, stacja dysków 1541, drukarkę Commodore MPS 802, 100 dyskietek [redacted] oprogramowaniem, dwa joysticki, literaturę. Cena kompletu 3,5 mln zł. Norbert Morajda, 33-170 Tuchów, ul. Kopernika 14, tel. 63-76.

Sprzedam C-64 II, stacja dyskow 1540 II (gwarancja), magnetofon, joystick, cartridge Black Box, oprogramowanie na kasetach, dyskietki, literaturę. Robert Dudziński 02-595 Warszawa, ul. Puławska 109B m 3.

Pilnie sprzedam C-64, magnetofon, dwa joysticki SV-119, cartridge Super Games, Final III, Black Box, oprogramowanie na kasetach, literaturę, mapę pamięci do C-64, C-64 - podręcznik użytkownika, instrukcję obsługi C-64. Cena kompletu 2,3 mln zł. Piotr Nowak, 40-832 Katowice, ul. W. Witosa 13C/18.

Sprzedam komplet: Amiga 500 2,5 [redacted] RAM, monitor Philips CM8833 II [redacted] filtrem, dodatkowa stacja 3,5", modulator TV520, około 350 dyskietek [redacted] oprogramowaniem, pudełka na dyskietki, literaturę. Marek Kul, 76-200 Słupsk, PO BOX 460, tel. 433-142

Sprzedam:
Amigę 500, rozszerzenie pamięci RAM do 2,3 - 6 mln
Monitor 1084 - 3,5 mln
Stację dysków - około 1 mln
Mysz, [redacted] pad Golden Image (gwarancja) - 320 tys.
Literaturę, dyskietki.
Krzysztof Kowalski, 41-800 Zabrze, ul. Nad Kanałem 34B/29, tel. (0-32) 75-34-93

Sprzedam C-128D, telewizor JUNOST, mysz, magnetofon, cartridge Final, dyskietki z oprogramowaniem, literaturę. Oferty z ceną proszę nadsyłać na adres; Tomasz Kontek, 22-100 Chełm, ul. Polaniecka 2/49

Sprzedam tanio C-64 stan idealny, magnetofon, Black Box, literaturę, oprogramowanie. Cena do ustalenia. Dariusz Orłowski, 63-400 Ostrów Wielkopolski, ul. Piłsudskiego 14A.

Sprzedam roczny C-64 IIC stan idealny. Komputer [redacted] z oprzyrządowaniem, programami i literaturą. Cena całosci 2,1 mln zł. Robert Kania, 64-920 Piła, ul. Roosevelta 72/5, tel. 13-16-80.

Sprzedam oryginalną dokumentację karty przyspieszającej 14 Mhz [redacted] Amigi 500. Pieniądze - [redacted] tys. zł. - proszę przesłać przekazem pocztowym na adres. Leszek Gieniec, 33-300 Nowy Sącz, ul. Lwowska 136/1.

Sprzedam lub zamienię na Amigę [redacted] następujący sprzęt: Commodore C-128, stację dyskow, magnetofon, dyskietki, kasety, oprogramowanie, literaturę, cartridge. Oferty proszę przesyłać na adres Witold Kaczor, 35-111 Rzeszów, ul. Lewakowskiego 5/33.

Sprzedam C-64 II, 1541 II (gwarancja), monitor Philips (zielony, gwarancja), magnetofon, Final III, X, joystick, pudełko na dyskietki, oporogramowanie, literaturę. Cena 6,570 mln zł. - możliwa negocjacja ceny. Piotr Tracz, 22-361 Żulin, woj. chełmskie.

Sprzedam numery BAJTKA z lat 1986-1992, kilkanaście sztuk COMMODORE [redacted] AMIGA i TOP SECRET. Rafał Walendowski, 56-400 Oleśnica Śląska, Spalice 39B.

Sprzedam C-64 II, magnetofon, telewizor, dwa joysticki SV 126, moduł, oprogramowanie i literaturę. Cena 3,1 mln zł. Sebastian Stroński, 46-250 Wołczyn, ul. Słowackiego 3A/9, tel. 419.

Sprzedam Commodore C-64 (gwarancja), magnetofon 1530, FINAL III, oporgramowanie i literaturę. Stan idealny, [redacted] 2,25 mln zł. Marcin Syzdoł, 26-060 Chęciny, Czerwona Góra 1/9, telefon 15-20-09.

Sprzedam moduł EX PLUS wraz z instrukcją [redacted] języku polskim. Cena urządzenia 50 tys. zł. Nawiążę kontakt [redacted] użytkownikami C-64 w celu nauki programowania [redacted] asemblerze, a także tworzenia gier. Wszystkich zainteresowanych proszę [redacted] kontakt osobisty lub listowny. Marek Letki, 40-570 Katowice ul. Ligocka 5/38.

Sprzedam C-128, magnetofon, moduły SUPER GAME, FINAL III, programy użytkowe, oprogramowanie, literaturę, telewizorek czarno-biały. Radosław Boroński, 62-510 Konin-Zatorze, ul. Karłowicza 3/27.

ZAMIENIĘ

Zamienię magnetofon 1530 do C-64, oprogramowanie 5 książek, dwa moduły, wkrętak do głowicy, grę elektroniczną, słuchawki do walk[redacted] na używaną i sprawnę stację dysków 1541 II. Dawid Podha... 59-220 Legnica, ul. Wańkowicza 1/3, tel. 61-792.

Zamienię na używaną stację dysków: magnetofon [redacted] C-64 (Datassette Unit 1530), moduł Black Box, dwa joysticki, oprogramowanie na kasetach. Informacja, koperta zwrotna [redacted] znaczkiem. Małgorzata Lewandowska, 09-305 Zieluń, ul. Szkolna 9.

Zamienię Amigę [redacted] na Amigę [redacted] peryferiami. Jakub Rachalewski, Zduńska Wola, telefon 28-84.

Za używaną stację dyskow do C-64 oferuję ciekawy sprzęt [redacted] gotowkę. Mariusz Mincewicz, 21-350 Międzyrzec Podlaski, ul. przedszkolna 7 m 24.

Zamienię motocykl Jawa 50 (nowy silnik) [redacted] częściami zapasowymi [redacted] Amigę [redacted] z osprzętem lub na Atari 1040 STE [redacted] monito[redacted] SM 124 lub [redacted] 144. Hubert Polański, Skarżysko-Kamienna, tel. 514-707.

Zamienię [redacted] C-64 (magnetofon, cartridge Black Box) następujący sprzęt: Atari 800 XL z magnetofonem [redacted] systemie Turbo 2000 + moduł, oprogramowanie na kasetach, literaturę. Wszystko [redacted] bardzo dobrym stanie. Grzegorz Janus, 56-120 Brzeg Dolny, ul. Tęczowa 8/18, tel. 192-112.

Zamienię Simsona Enduro [redacted] kaskiem i owiewkami [redacted] Amigę 500 [redacted] monitorem lub modulatorem TV. Mariusz Korż, 76-200 Słupsk, Al. 3-go Maja 80/124, tel. 43-63-85.

RÓŻNE

Proszę [redacted] kontakt posiadaczy licencjonowanej wersji Lemingów na C 64. Tomasz Tracikowski, 33-300 Nowy Sącz, ul. Nawojowska 21/19, [redacted] 216-64.

Poszukuję literatury: „Asembler 6502" Janusza Ruszczyca, „Jak rozbudować interpreter C-64" K. Gajewskiego, B. Radziszewskiego oraz dobrej mapy pamięci C-64. Nawiążę kontakt [redacted] programującymi [redacted] asemblerze. Maciej Zieliński, 37-310 Nowa Sarzyna, ul. Bolesława Prusa 2/35.

Nawiążę kontakt [redacted] osobami uczącymi się asemblera (stacja dyskow 1541 II). Poszukuję schematu samplera do C-64. Tomasz Wójcik, 30-382 Kraków, ul. Kobierzyńska 112.

Zlecę napisanie programu na Amigę 500. Oferty kierować: Marek Trzęsicki, ul. 22 Lipca 7, 46-243 Bogacica.

Grupa FATUM (C-64) poszukuje muzyków, grafików, koderów. Kontakt (kaseta/dysk): Sławomir Petrykowski, 11-200 Bartoszyce, ul. Nad Łyną 3/28, tel. 33-00.

Poszukuję modułu [redacted] interpreterem SIMONS BASIC [redacted] literaturę po polsku na temat tego języka. Artur Kawa, 42-300 Myszków, ul. Kościuszki 68/70 m 3.

IMPOSSIBLE MISSION 2



Co za podły świat. Nie dość, że byłeś tego dnia w złym nastroju, to jeszcze ten facet. Wiecznie pętałeś się gdzie nie potrzeba i z wciąż brzmiącymi w uszach słowami: „JESZCZE JEDEN GOŚĆ... ZOSTANIESZ TU NA ZAWSZE!".

Znalazłeś się... no właśnie, gdzie? Wygląda to na jakiś kompleks podziemnych budynków obwarowanych przez najróżniejsze cuda techniczne - pętające się gdzieniegdzie roboty, podglądające kamery, windy, pozamykane przejścia. Nie chcę cię martwić, ale wpakowałeś się w sytuację bez wyjścia.

Zaraz, zaraz, przecież kilka lat temu byłeś w podobnej sytuacji. Także jakiś nieprzyjazny osobnik zamknął cię w całkiem podobnie wyglądającym więzieniu, ale to było tak dawno, w IMPOSSIBLE MISSION 1. Dzięki wrodzonej spostrzegawczości udało ci się jakoś opuścić niegościnne progi podziemnych budowli.

Teraz sytuacja wygląda podobnie, bo - nawet jeszcze gorzej. Robotów jest jeszcze więcej, kamery i tym podobne gadżety są tu prawie na każdym kroku. A więc idź i szukaj potrzebnych ci przedmiotów w kanapach, zestawach stereofonicznych, lustrach. Ale musisz wykazać się niesamowitą wprost zręcznością, żeby nie wpaść w jakąś zakazaną dziurę, nie dać się zestrzelić robotowi, czy wreszcie nie wykonać innego równie głupiego posunięcia. Nie jest to proste. Nieostrożny gracz zginie tutaj przeciętnie raz na sekundę, a więc uważaj. A poza tym - spiesz się, bo czasu na wydostanie się jest mało.

IMPOSSIBLE MISSION 2 ma w sobie coś takiego, że potrafi nawet najbardziej znudzonego gracza wciągnąć na długie godziny w szalone przygody bohatera gry - działa jak magnes. Powodem tego nie jest grafika, która plasuje się mniej więcej na poziomie C-64 (no, może troszeczkę lepiej), ani dźwięk (jakbym słyszał komodorka...), a po prostu sama idea gry, fakt, że przemieniasz się w tajnego agenta usiłującego wydostać się z twierdzy wroga.

Trzeba przyznać, że nie na darmo na opakowaniu znajduje się napis CLASSIC - gra ta rzeczywiście należy już do klasyki. Niegdyś zagrywali się w nią komodorowcy, teraz - amigowcy. IMPOSSIBLE MISSION 2 może zaskoczyć jednym - okrzykami przerażenia (generowanymi przez komputer) brzmiącymi niezwykle realistycznie. Życzę wam, żeby tych okrzyków było jak najmniej.

VOYAGER

FIRMA: EPYX
RODZAJ GRY: zręcznościowo-przygodowa
KOMPUTER: Amiga
WYMAGANIA: -

GRAFIKA
MUZYKA
OGÓLNIE

BALON

Ta gra jest również grą zręcznościową autorstwa Sławomira Szczyńskiego. Program ten dostępny jest zarówno na taśmie jak i na dyskietce u jedynego jej dystrybutora - Fundacji Edukacji Technologicznej.

Zabawa polega na pilotowaniu balonu. Należy dobrze się skupić, by podczas wykonywania manewrów nie zostać zestrzelonym przez naziemnych wrogów lub nie zderzyć się z przelatującymi ptakami. Nasz balon nie może wylądować na ziemi, gdyż spowodowałoby to utratę jednego z czterech żyć. Do naziemnych obiektów, które znajdują się w ciągłym ruchu, możemy strzelać. Naciskając przycisk FIRE (joystick 2) rzucamy na wroga bomby. Częstotliwość zrzutów uzależniona jest od wysokości, na jakiej znajduje się balon. Po wyrzuceniu trzech bomb trzeba poczekać by eksplodowały, a dopiero potem można wyrzucić następne.

A teraz kilka wskazówek.

ETAP 1: Nasza podróż rozpoczyna się na ziemi i jest ograniczona limitem czasu. Wyświetlacz czasu znajduje się w prawym górnym rogu ekranu. Naziemni wrogowie znajdują się w wozach ciągniętych przez konie. Z wozów tych wystrzeliwane są pionowo w górę rakiety. Ptaki wylatują z prawej strony ekranu na takiej wysokości, na jakiej znajduje się w danej chwili nasz balon. Ptaka można zestrzelić. Możliwe jest to tylko wtedy, gdy wzniesiemy się nad niego i rzucimy bomby.

W całej grze obowiązuje następująca punktacja: za zestrzelenie ptaka - 200 punktów, za zestrzelenie naziemnego celu - 100 punktów. Ponadto ukończenie każdego z etapów wiąże się z otrzymaniem premii - 1000 punktów. Po zdobyciu 20, 50 i 70 tysięcy punktów otrzymujesz dodatkowe życie.

ETAP 2: Zmieniają się tu naziemni wrogowie i broń, z której do nas strzelają.

SHANGHAI



Gra ta jest stara jak świat dosłownie i w przenośni, bowiem wymyślono ją w starożytnych Chinach. Do gry w SHANGHAI potrzebna Ci będzie dobra pamięć (oczywiście Twoja, a nie komputera), szczęście i umiejętność strategicznego rozumowania. Układanka ta składa się ze 144 płytek ułożonych w stos zwany „Smokiem". Rozgrywka polega na zdjęciu ze stosu wszystkich płytek, jednak nie w sposób dowolny. Należy zdejmować pary płytek tak długo, aż nie będzie można wykonać żadnego ruchu. Dozwolone jest zdejmowanie ze stosu tylko „wolnych" płytek, tzn. takich, które nie są otoczone jednocześnie z prawej i z lewej strony przez inne płytki oraz nie znajdują się pod innymi płytkami. Oczywiście wszystko to dotyczy tylko par identycznych płytek (poza tymi, które określają kwiaty i pory roku). A kiedy uda Ci się opróżnić cały stół, czeka Cię nagroda. Ostrzegam jednak: zwycięstwo nie jest proste.

Teraz kilka słów na temat strategii gry. Skoncentruj się na czterech płytkach blokujących główne linie ruchu: jednej wysuniętej najbardziej na lewo, dwóch wysuniętych najbardziej na prawo i tej, która leży na samym szczycie. Najważniejsze jest rozkładanie rzędów, a później redukowanie wysokości stosu. Uważaj, gdy trafisz na odsłoniętą trójkę płytek (z czterech możliwych). Wtedy zastanów się dobrze, jakie posunięcie może być dla Ciebie najbardziej korzystne. Natomiast kiedy będziesz miał „wolne" wszystkie cztery płytki jednego rodzaju - bez zastanowienia zdejmij je wszystkie. Staraj się też planować swoje posunięcia kilka ruchów naprzód.

Oprócz normalnej zabawy w zdejmowanie płytek można ustalać limity czasowe dla całej rozgrywki oraz dla pojedynczego ruchu. Możliwa jest też rozgrywka dla większej ilości osób (od 1 do 5). Gdy stwierdzisz, że Twój ruch był niewłaściwy, możesz go cofnąć (nawet do samego początku gry). A kiedy nie wiesz, jaki wykonać następny ruch, spojrzyj do ściągawki - komputer pokaże wszystkie możliwe w danym momencie posunięcia.

Na dysku wraz z grą znajdują się przykładowe rozłożenia, którymi można się pobawić. A jeśli spodoba Ci się jakiś układ, w który grałeś, zawsze możesz go sobie zapisać go na dyskietce.

SHANGHAI bynajmniej nie zwala z nóg doskonałą grafiką i dźwiękiem. Pod tymi względami plasuje się raczej na przeciętnej pozycji. Ale jako gra logiczna, jest po prostu niezastąpiony. To jedna z najciekawszych gier logicznych, jakie do tej pory widziałem.

VOYAGER

FIRMA: Activision
RODZAJ GRY: logiczno-strategiczna
KOMPUTER: Amiga
WYMAGANIA: -

GRAFIKA
MUZYKA
OGÓLNIE

GRY • GRY • GRY

ETAP 3 Teraz przelatujemy nad wodą. Na jej powierzchni, na tratwach pływają wrogowie rzucając [redacted] nas czerwonymi kulkami. Ptaki są bardziej kolorowe a ich tor lotu przypomina wynik badania EKG.

ETAP 4. [redacted] tym etapie nie używamy przycisku FIRE (brak wrogów). Należy jedynie omijać trąby powietrzne.

ETAP 5: Ponownie pojawią się ptaki i wrogowie, którzy strzelają do nas [redacted] swych rydwanów. Co do ptaków, to chcę zaznaczyć, iż wysokość ich lotu jest bezpośrednio związana [redacted] położeniem balonu. O ile [redacted] poprzednich etapach ptaki latały torem [redacted] góry zaprogramowanym, o tyle [redacted] tym etapie dążą do uzyskania identycznego położenia jak balon.

ETAP 6 To już prawie koniec naszych zmagań. Trzeba jeszcze tylko wylądować w miejscu, w którym znajduje się migająca strzałka. Uwaga na mury zamku i latające nad nim ptaki!

Po ukończeniu szóstego etapu dostajemy premię [redacted] wysokości 5000 punktów [redacted] komputer powraca do pierwszego poziomu. W grze następują zmiany, które mają utrudnić nam ponowny lot. Zwiększa się liczba ptaków i naziemnych wrogów, a trąby powietrzne poruszają się szybciej.

Krótko mówiąc, jeśli chcesz dobrze się zabawić - zagraj [redacted] Balona

ROBERT KULIŚ

[redacted] Sławomir Skrzyński
[redacted]
Fundacja Edukacji Technologicznej, [redacted]
[redacted] zręcznościowa
[redacted] C-64
[redacted] -

GRAFIKA
MUZYKA
OGÓLNIE

KANGOROO

Oto gra zręcznościowa, której autorem jest pan Sławomir Skrzyński z Rypina. Grę tę można nabyć zarówno na kasecie, jak i na dyskietce za pośrednictwem jedynego jej i dystrybutora - Fundacji Edukacji Technologicznej [redacted] Warszawie.

Bohaterami gierki są dwa kangury. Twoim zadanie polega [redacted] takim pokierowaniu kangurem koloru brązowego, by spotkał się on [redacted] drugim koloru żółtego. Wykonanie tego zadania nie jest jednak takie proste, jak mogłoby się wydawać.

Gra składa się [redacted] czterech plansz. Każda z plansz zbudowana jest z kilku pułapów połączonych ze sobą stopniami i drabinkami. Na każdym [redacted] pięter znajdują się nasi wrogowie-małpy. Są to, jak wiesz, dość wredne stworzenia, więc nie zdziw się, jeśli zostaniesz trafiony jakimś ogryzkiem lub jabłkiem. Możesz tego oczywiście uniknąć np. podskakując lub kładąc się (drążek joysticka do przodu lub do tyłu).

Małpy pojawiają się [redacted] po prawej stronie ekranu. [redacted] czasie gdy schodzą z drzewa, masz jedyną szansę unieszkodliwienia którejś z nich. Musisz podejść do drzewa i nacisnąć przycisk FIRE. Twój kangur wyciągnie łapę i dotknie małpy przez co [redacted] pewien ją unieszkodliwi.

[redacted] czasie wędrówki napotkasz owoce. Za zbieranie ich otrzymujesz punkty. Jeśli podskoczysz i uderzysz [redacted] dzwon, to pojawią się nowe owoce. Właśnie [redacted] ten sposób można zarobić wiele punktów. W grze masz do dyspozycji ograniczoną ilość czasu, którego wskaźnik znajduje się w lewym górnym rogu ekranu. Również i ilość żyć jest ograniczona do trzech.

Życzę powodzenia i przyjemnej zabawy. Niechaj kangury się spotkają, a małpy poskręcają - [redacted] co!

ROBERT KULIŚ

[redacted] Sławomir Skrzyński
[redacted]
Fundacja Edukacji Technologicznej, [redacted]
RODZAJ GRY: zręcznościowa
KOMPUTER: C-64
WYMAGANIA: -

GRAFIKA
MUZYKA
OGÓLNIE

GARFIELD WINTER'S TAIL

Znacie kota Garfielda? Charakteryzuje [redacted] to, że dużo je, bardzo dużo śpi i dużo myśli. Jest straszliwie zabawny, otyły, no i wciąż chodzi niewyspany. Komiksowa postać kota stworzonego przez Jima Davisa fascynuje zarówno dzieci, jak i dorosłych. Ten kot jest po prostu tak zabawny, że nawet największego ponuraka przyprawi o uśmiech swoimi ciągłymi utarczkami [redacted] psem Oddie i swoim zakompleksionym panem Johnem. Oddie codziennie skacze jak oszalały, bo zapomniano nauczyć go chodzić, John poszukuje wciąż wybranki swojego serca (ostatnio była to kobie[redacted] brodą), a Garfield, jeśli nie śpi, to właśnie je swój kolejny, dzielący dżemki posiłek.

O domu, [redacted] którym mieszka Garfield, na pewno nie można powiedzieć, że jest "normalny". Tam zawsze coś się dzieje i przeważnie przegranymi są John i Oddie, a zwycięzcą Garfield ze swoim kocim sprytem.

Kot ten stał się już inspiracją do wielu kreskówek, jego maskotkę można kupić w każdym sklepie [redacted] zabawkami w USA. Tym razem Garfield zawędrował [redacted] krainę gier. Być może znacie [redacted] już lub widzieliście w głównej roli gry Garfield na C-64, teraz możecie go ujrzeć przed sobą pośród zawieji śnieżnej.

GARFIELD WINTER'S TAIL to kolorowa i zabawna gra zręcznościowa dla dzieci. Składa się [redacted] trzech części. [redacted] pierwszej Garfield [redacted] Oddim próbują swoich sił [redacted] zjeździe [redacted] nartach. Trasa nie jest zbyt łatwa, obfituje w wiele przeszkód i Wasz kotek nie raz będzie miał zderzenie czołowe [redacted] śniegiem. Nie dajcie się wyprzedzić Oddiemu. Po drodze, jeśli chcecie uzupełnić energię, musicie wykradać ciasteczka i inne pokarmy, które nierozważny John postanowił skonsumować na trasie zjazdu.

[redacted] drugiej części Garfield wybrał się na łyżwy. Musi jeździć po pozostawionych odciskach stóp jakiejś kury. To też nie jest proste, bo szybkość jest zawrotna, [redacted] o wywrotkę nietrudno. Uważajcie na rzekę, bo bardzo łatwo [redacted] niej wylądować. Nie zatrzymujcie się także nazbyt długo [redacted] jednym miejscu, bo już po chwili pojawi się Oddie i otoczy Was przeręblami.

Akcja trzeciej części rozgrywa się na kurzej farmie. Przed Garfieldem stoi trudne zadanie. Musi każdej kurze poprzez skomplikowany sytem rur dostarczyć karmę. Kiedy mu się to uda, otworzą się drzwi do zamkniętej windy. Po drodze, [redacted] ramach relaksu można dołożyć kopniaka Oddiemu (widzicie, jak fajnie poleciał?). Wszystko to oczywiście dzieje się [redacted] snach naszego kotka, bo [redacted] to, żeby wybrać się [redacted] narty Garfield jest zbyt leniwy...

Dźwięk i grafika nie są rewelacyjne, jednak gra ta [redacted] pewnością przyciągnie na długo uwagę najmłodszych graczy.

VOYAGER

[redacted] THE EDGE
[redacted] zręcznościowa
[redacted] Amiga
[redacted] -

GRAFIKA
MUZYKA
OGÓLNIE

W następnym numerze :

- ***Gry, gry i jeszcze raz gry !***
- ***Wojskowe zastosowania komputerów***
- ***Krzyżówka z nagrodami dla maniaków amigowskich gier***
- ***C-64 - sterowanie robotami***
- ***Historia sceny C-64***

GRY • GRY • GRY

JAK NAPISAĆ WŁASNE DEMO (CZ. 6)

Fonty, Loga, Rysunki... odsłona 2

W poprzednim odcinku zajmowaliśmy się grafiką znakową, nauczyliśmy się wyświetlać teksty, zmieniać obszar położenia mapy i generatora znaków. Nie jest to jednak wszystko, co dotyczy grafiki znakowej. W tym arykule postaram się wyczerpać ten temat dogłębnie, a także zapoznam Was z grafiką bitmapową.

Extended Color Interpretation

Bardzo ważne z naszego punktu widzenia jest tak zwane ECI (Extended Color Interpretation - rozszerzona interpretacja koloru, albo - jak spotyka się w niektórych publikacjach - Extendend Background Color Mode). Jak już dowiedzieliśmy się wcześniej, ekran znakowy może mieć tylko jeden wspólny kolor, za który odpowiada komórka $d021 (53281). Istnieje możliwość zwiększenia ilości kolorów tła (!), lecz wiąże się to z ograniczeniem generatora znaków do 64 pozycji (kody znaków od $0 do $3f (63)).

Jak to działa? Otóż za włączenie trybu ECI odpowiedzialny jest 6 bit rejestru $d011. W momencie, w którym jego wartość wynosi 1, komputer tworząc ekran postępuje trochę inaczej niż zwykle. Bajt z pamięci ustawienia znaków jest traktowany w następujący sposób: sześć młodszych bitów odpowiada za numer znaku (26 = 64, czyli możemy korzystać z sześćdziesięciu pierwszych znaków generatora. Nie jest to wiele ale wystarcza na cały alfabet, znaki przestankowe i cyfry. Natomiast dwa najstarsze bity (nr sześć i siedem) odpowiadają za kolor tła, na jakim ma być wyświetlany dany znak. Za kolory teł znaków odpowiadają komórki $d021, $d022, $d023 i $d024. Zależność ustawienia bitów przedstawia tabela 1.

Tabela 1

Zależność komórki odpowiedzialnej za kolor od ustawienia odpowiednich bitów w pamięci ustawienia znaków.

Bit 7	Bit 6	Kolor
0	0	$d021
0	1	$d022
1	0	$d023
1	1	$d024

Aby zobaczyć jak to działa, wpisz następujący program (oczywiście w turboasemblerze):

```
*= $1000
lda $d011
ora #$40
sta $d011
lda #[illegible]
sta [illegible]
sta $d021
lda #[illegible]
jsr $[illegible]d2
lda #$0d
jsr $ffd2
ldx #$02
stx $d022
inx
stx $d023
inx
stx $d024
lda #$01
sta $0400
lda #$41
sta $0401
lda #$81
sta $0402
lda #$c1
sta $0403
rts
```

Program ten wypisuje w lewym górnym rogu ekranu cztery literki „A", z których każda jest na innym tle. Możemy teraz poeksperymentować zmieniając zawartość komórek odpowiedzialnych za kolor i powypisywać znaki o różnych kodach. Prawda, że proste?

Ciekawy efekt można uzyskać definiując zanki w rewersie (czyli tło zaznaczamy jako 1, a kształt litery jako zero). Teraz, gdy włączymy tryb ECI, zamiast teł kolory pozmieniają same znaki.

No tak, ale do czego może się nam to przydać? Widziałeś już na pewno jakiś magazyn dyskowy (np. nie istniejący już HIGHLIFE grupy PARADOS). W tekstach, które można przesuwać góra/dół są podświetlane, migające napisy. Były one właśnie zrealizowane techniką ECI. Technika ta jest w tym przypadku szczególnie przydatna, gdyż wystarczy zmienić zawartość jednej komórki by zmienił się kolor wszystkich napisow, za które jest ona odpowiedzialna. Jest to znacznie prostsze niż posługiwanie się zmianami w pamięci koloru, które wymagałyby skomplikowanych operacji przeliczania adresów przy równoczesnym analizowaniu danych znajdujących się aktualnie na ekranie. Jeżeli chcesz przekonać się jak to działa, wpisz program z listingu 1.

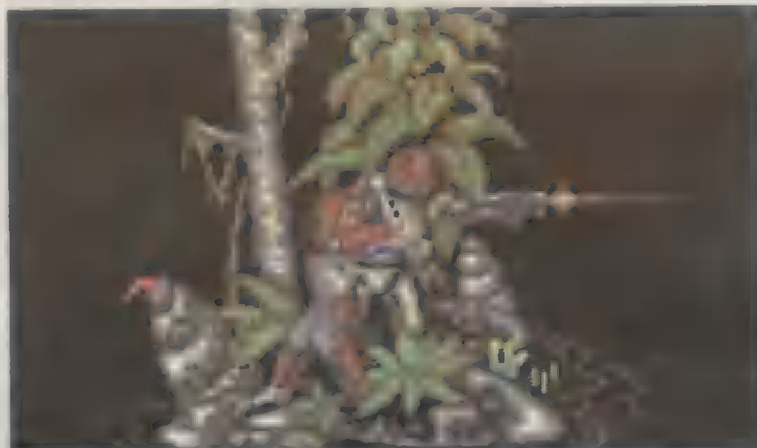
Multicolor - znaki wielokolorowe

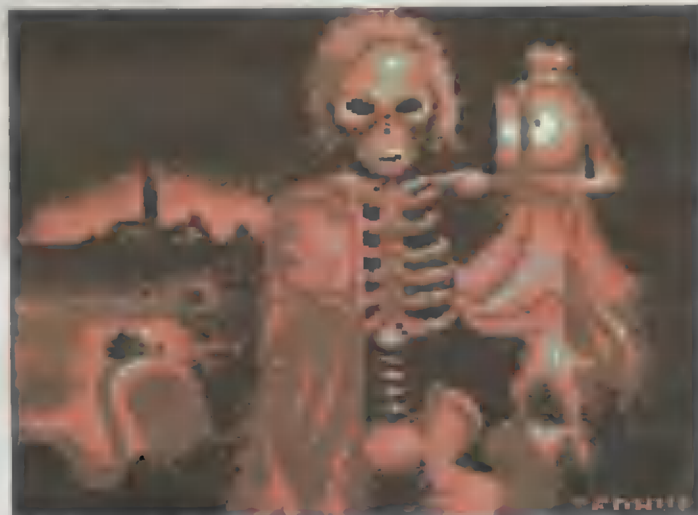
Kolejną sprawą jest tryb wielokolorowy (ang. multicolor mode). Wywołuje się go zapalając czwarty bit w komórce $d016. W tym trybie rozdzielczość jest dwa razy mniejsza, ale za to można uzyskać cztery kolory w jednym znaku, bo za kolor punktu odpowiadają dwa sąsiednie bity. Jak to się właściwie dzieje?

Otóż trzeba wiedzieć, że samo włączenie multicoloru nie spowoduje ukazania się tego trybu. Dlaczego? Bo komputer zaczyna w nieco inny sposób interpretować pamięć koloru ($d800). Jeżeli kolor danego ($d800) znaku jest z przedziału 0-7, to komputer wyświetli ten znak w podanym kolorze w wysokiej rozdzielczości. Natomiast jeżeli kolor znaku jest z przedziału 8-15, to komputer wyświetli go w trybie multicolor, ale za to w kolorze „o 8 mniejszym" niż zawartość nibbla koloru. Inaczej mówiąc za kolor znaku odpowiadają bity od 0 do 2, a za tryb muticolor - bit 3.

Po co to jest potrzebne? A na przykład po to, by wyświetlić ładną kolorową ramkę, a w środku niej wyraźny napis w wysokiej rozdzielczości.

Za to, jakie będą pozostałe kolory, odpowiadają komórki $d022 i $d023. Jak to się ma do ustawienia bitów, przedstawia tabela 2:





Dema prezentowane na szczecińskim copy party w grudniu ub. r.

Tabela 2

Komórki pamięci odpowiedzialne za przydział kolorów w trybie znakowym multicolor	
Bity	Skąd brac kolor
00	$d021 - kolor tła
01	$d022 - pierwszy kolor dodatkowy
10	$d023 - drugi kolor dodatkowy
11	nibble pamięci koloru ($d800)

Tryb multicolor-tekst używa się najczęściej do rysowania logosów do intr, gdyż można uzyskac stosunkowo niezłe efekty przy niewielkim zużyciu pamięci, a także w dość prosty sposob wykonywać na nich operacje, które powodują powstawanie ciekawych efektów wizualnych (np. tech-tech, display list itp.). Ostatnio odchodzi się jednak od tego i umieszcza rysunki w trybie graficznym, gdyż dają one grafikowi o wiele większe pole do popisu, a problem zajętosci pamięci nie jest już tak palący ze względu na to, iż wciąż powstają nowe, coraz lepsze pakery.

Właściwie był to ostatni z trybów tekstowych, myślę zatem, że teraz omówić

Tryby graficzne

Różnice między trybem graficznym a tekstowym są niewielkie. Po prostu ekran graficzny jest zorganizowany jakby był mapą składającą się z tysiąca znaków (40x25) z tym, że nie ma jawnej mapy ich ustawienia, tylko poszczególne wyświetlane są w kolejności.

Taka organizacja ma pewną przewagę nad organizacją liniową. Po pierwsze łatwiej jest wyświetlać teksty w trybie graficznym. Po drugie, mimo że wyliczenie, który piksel zapalić, jest bardziej skomplikowane, to przy zastosowaniu kilku prostych tabelek tryb graficzny pozwala na napisanie procedury kreślenia punktu o kilka cykli szybszej niż byłoby to w przypadku organizacji liniowej. Ogólny wzór pozwalający nam wyznaczyć, który bit w którym bajcie należy zapalić, przedstawia się w następujący sposób:

adres=320*int(y/8)+y-8*int(y/8)+8*int(x/8)+peg
bit=7-(x-8*int(x/8))

gdzie:
x,y - współrzędne punktu,
peg - początek ekranu graficznego.

No dobrze, znamy już organizację ekranu, teraz pora dowiedzieć się, jak uzyskać tryb graficzny. Odpowiedzialny za to jest piąty bit komórki $d011 (53265), ustawiając go na jeden włączamy tryb graficzny. Jednocześnie pamięć ustawienia znakow jest od tej pory traktowana jako pamięć kolorów ($d800 jest teraz nie używane). Wynika z tego, że dla każdego znaku możemy ustawić oddzielne atrybuty (kolor tła i atramentu), przy czym za bity ustawione na jeden odpowiada starszy nibbel, a za bity ustawione na zero - młodszy nibbel.

Ekran graficzny zajmuje prawie 8 KB (dokładniej 8000 bajtów), możliwe są więc tylko dwa ustawienia w banku VIC-a: na jego początku lub końcu. Wyboru dokonuje się przy pomocy bitu zerowego komorki $d018 (53272).

Cały czas była tu mowa o grafice w trybie Hi-Res (320x200), ale możemy też wyświetlać grafikę w trybie multicolor, ustawiając bit czwarty rejestru $d016. Tutaj panują jednak trochę inne zasady niż przy trybie tekstowym. Przede wszystkim nie możemy mieszać sposobów wyświetlania. Cały ekran wyświetlany jest albo w trybie multicolor, albo w Hi-Res (chyba, że podziałamy trochę przerwaniami, lecz to zupełnie inna historia). Aktywna jest też pamięć koloru ($d800), co pozwala na nadanie każdemu znakowi trzech niezależnych atrybutów. Tabela 3 przedstawia zależność kolorów od ustawienia bitów:

Tabela 3

Komórki pamięci odpowiedzialne za przydział kolorów w trybie graficznym multicolor	
Bity	Skąd brac kolor
00	$d020 - kolor tła wspólny dla całego ekranu
01	starszy nibbel pamięci ustawienia znaków (kolorow)
10	młodszy nibbel pamięci ustawienia znaków (kolorów)
11	młodszy nibbel pamięci kolorów ($d800)

Są to już wszystkie tryby graficzne, jakie oferuje nam nasz C-64, ale... No właśnie, podczas pisania demek udało się co poniektórym wzbogacić komodorka w sposób programowy o kilka dodatkowych trybów. Nie będę ich tutaj dokładnie omawiał, ponieważ zajęłoby to zbyt wiele miejsca, a poza tym jesteśmy jeszcze na to zbyt mało przygotowani, gdyż uzyskanie tych trybów wymaga bardzo zaawansowanego cyklowania. W przyszłości, jeżeli wykażecie zainteresowanie, omówimy je dokładniej. Na razie chciałbym tylko zasygnalizować ich istnienie. I tak:

FLI (Flexible Line Interpretation) jest to tryb, wynaleziony przez holenderską grupę BLACK MAIL. Podobny do trybu multicolor, z tym że mamy możliwość użycia większej liczby kolorów dla danego znaku. Uzyskuje się to poprzez zmianę co linię rastra adresu pamięci ustawienia znaków. Niestety nie można wykorzystywać trzech pierwszych kolumn znakowych. Obrazek w trybie FLI zajmuje dużo pamięci (17 KB) i pochłania czas procesora, który jest zajęty przez tyle linii, przez ile wyświetlany jest tryb FLI, za to efekty mogą być oszałamiające. Dokładniejszy opis tego trybu możecie znaleźć w „C&A" 11/92. Programy graficzne wykorzystujące ten tryb: FLI EDITOR, FLI-GRAPH, INTERPAINT.

A-FLI (Advanced Flexible Line Interpretation). Jest to tryb wysokiej rozdzielczości uzyskiwany w sposób bardzo podobny do FLI. W trybie tym każdy bajt mapy bitowej ma własne atrybuty i, podobnie jak we FLI, trzy pierwsze kolumny znakowe nie nadają się do użytku. Wymyślony został przez D'ARCa z grupy TOPAZ BEERLINE. Obrazek AFLI zajmuje 16 KB. Programy graficzne wykorzystujące ten tryb: AFLI-DESIGNER.

INTERLACE to tryb podpatrzony na Amidze. Polega na tym, że bardzo szybko (50 razy na sekundę) wymienia się wyświetlane ekrany, co oprócz nieznacznego migotania powoduje złudzenie (zależnie od rodzaju rysunków) albo zwiększenia rozdzielczości, albo liczby kolorów. Jednak obraz taki jest szkodliwy i męczący dla oczu. Programy graficzne wykorzystujące tryb INTERLACE: INTERPAINT.

Jeszcze innym trybem jest, na razie jeszcze przez nikogo nie zrealizowany, choć wielu myśli o tym od dawna, tryb Super-HiRes, polegający na nałożeniu na ekran wysokiej rozdzielczości dwóch warstw sprajtów. Dzięki tej metodzie możemy stworzyć obrazek mający rozdzielczość 96x200 punktów, w czterech kolorach, przy rozmiarze pikseli takim jak w trybie Hi-Res. Prawdopodobnie pierwszą osobą która użyje tego trybu w demie, będzie Polonus z grupy PADUA (fragment jego niedokończonej części mogli zobaczyc uczestnicy copy party w Szczecinie w grudniu 1992 roku). Programy graficzne wykorzystujące ten tryb: brak.

Ciekawym pomysłem może byc skrzyżowanie FLI i INTERLACE'u. W ten sposób uzyskuje się dużą rozdzielczość i jednocześnie wiele kolorów. Przykładem dzieła w tym trybie jest rysunek z pierwszego artykułu naszego cyklu, przedstawiający Samanthę Fox. Programy graficzne wykorzystujące ten tryb: brak.

To by było na tyle, w następnym odcinku omówimy sobie wszystkie sprawy związane z wyświetlaniem sprajtów. Do zobaczenia.

(cdn.)

RAFAŁ PIASEK

LISTING 1

```
;-----------------------------
        *= $1000
;-----------------------------
init    sei
        jsr $ff5b ;inicjalizacja VIC
        ldx #>irq ;inicjalizacja
        ldy #<irq ;przerwan IRQ
        sty $0314
        stx $0315
        ldx #$5b
        stx $d011
        ldx #$f1
        stx $d01a
        ldx #$7f
        stx $dc0d
        ldx #$00
        stx $d00e
        ldx #$30
        stx $d012
        lda #$00
        sta $d020       ;ustawienie odpo-
        tax             ;wiednich kolorow
        sta $d800,x
        sta $d900,x
        sta $da00,x
        sta $db00,x
        inx
        bne *-13
        sta $0286
        lda #$[illegible]
        sta $d[illegible]1
;-----------------------------
        ldx #$0[illegible]  ;wypisanie tekstu
        lda text[illegible]
        sta $040[illegible]
        inx
        cpx #$28
        bne *-9
        lda $01         ;przepisanie
        pha             ;generatora znakow
        lda #$01        ;w rewersie
        sta $01
        ldx #$00
        lda $d400,x
        sta $2[illegible]00,x
        lda $d[illegible]0,x
        sta $2100,x
        inx
        bne *-13
        pla
        sta $01
        lda #$[illegible]9     ;zmiana generatore
        sta $d018
        cli
        jmp *
;-----------------------------
irq     ldx poz1        ;procedury zmie-
        lda kolorki,x ;niajace kolory
        sta $d022
        inx
        cpx #28
        bne *+4
        ldx #$00
        stx poz1
        ldx poz2
        lda kolorki,x
        sta $d[illegible]3
        inx
        cpx #28
        bne *+4
        ldx #$[illegible]
        stx poz[illegible]
        ldx poz[illegible]
        lda kolorki,x
        sta $d[illegible]24
        inx
        cpx #2[illegible]
        bne *+[illegible]
        ldx #$[illegible]
        stx poz3
        lda [illegible]1
        sta $d019
        jmp $ea31
;-----------------------------
poz[illegible]    [illegible]
poz2    [illegible]
[illegible]    .byte [illegible]4
;-----------------------------
kolorki .byte $06,[illegible]4
        .byte $0e,[illegible]5,$[illegible]
        .byte [illegible],$[illegible]
        .byte [illegible]5,$0[illegible],[illegible]
        .byte [illegible]2
;-----------------------------
text    .byte [illegible]$20[illegible]
        .byte [illegible],$4[illegible]
        .byte [illegible]$44,$4[illegible]
        .byte [illegible]$8e,$[illegible]
        .byte [illegible]
        .byte [illegible]$20[illegible]
        .byte [illegible]
;-----------------------------
```

KĄCIK POCZĄTKUJĄCEGO (cz. 6)

POWTÓRKA Z MATEMATYKI I NIEŚMIERTELNA INSTRUKCJA

Cześć! Od tego odcinka zaczynam się do Was zwracać „O, Prześwietni", bo wiecie już naprawdę sporo (o ile oczywiście czytaliście wszystkie odcinki cyklu) i byłaby to wielka obraza tytułować Was inaczej.

A zatem, o Prześwietni, wyznajecie się w bitach, bajtach, a nawet w magistralach danych. Nie pozostaje mi więc nic innego, jak tylko ugruntować Waszą wiedzę z dziedziny liczb binarnych (tj. tych w systemie, czy też układzie dwójkowym), bo poprzednio wyjaśniłem tę sprawę nieco zbyt lakonicznie - ważniejsze było sedno zupełnie innego problemu.

Zacznijmy więc od systemu... dziesiętnego. Rozłóżmy sobie liczbę np. 147 na czynniki pierwsze:

$147 = 1*10^2 + 4*10^1 + 7*10^0$

Jak widać, każda liczba składa się po prostu z kolejnych potęg podstawy systemu czyli dziesiątki (poczynając od potęgi zerowej - pierwsza cyfra od prawej), pomnożonych przez kolejne cyfry liczby. I to właśnie jest podstawowa zasada zapisu wszelkich liczb we wszelkich systemach - jeżeli ją zrozumiecie, stanie się dla Was jasny każdy zapis liczbowy, choćby i w systemie stuczterdziestosiódemkowym.

> **Skąd my to znamy?**
>
> - Każda liczba różna od zera podniesiona do potęgi zerowej równa się jeden.
> - Wynikiem podniesienia liczby do potęgi pierwszej jest ta sama liczba.

Liczba, którą potęgujemy, to podstawa systemu. W powyższym przykładzie jest to 10, czyli mamy do czynienia z systemem dziesiątkowym. W systemie dwójkowym podstawą jest... tak! Oczywiście dwa! Weźmy więc teraz na warsztat liczbę binarną 1001:

$1001 = 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0 = 9$ (dziesiętnie)

No, sami widzicie, że to wcale nie takie trudne.

A teraz, jako ćwiczenie podam Wam kilka liczb binarnych żywcem wyjętych z pamięci komodorka, bo zapisanych ośmioma cyframi czyli binarnie. Pamiętacie poprzedni odcinek? No właśnie, takie ośmiocyfrowe liczby dwójkowe są po prostu przedstawieniem zawartości jednego bajta. Na przyszłość radzę Wam wyobrażać sobie liczby „tkwiące w bajtach" właśnie w taki sposób, a nie w postaci liczby dziesiętnej, dzięki czemu będziecie mieli ułatwione zadanie podczas samodzielnego programowania. Ale do rzeczy.

Podam Wam kilka liczb binarnych, a Wy spróbujcie je sami przekonwertować na system dziesiętny. W razie czego (lub dla sprawdzenia) poprawnie obliczone liczby dziesiętne znajdziecie w ramce na tej samej stronie. A oto liczby dwójkowe „do przemyślenia":

10000000
00101011
11110000
10001000
01010101

Chyba wystarczy, prawda?

No, to dajmy sobie już spokój z systemem dwójkowym i przejdźmy do praktyki. Jak już wiecie, Wasz C-64 bezpośrednio po włączeniu dysponuje pewnym zasobem wiedzy i nie jest wcale taki głupi. Ten „zasób wiedzy" to wbudowany (ładniej i bardziej fachowo mówi się: zaimplementowany) interpreter języka BASIC. On to właśnie dba o to, by wyświetlić na początku dobrze znany Wam napis „**** COMMODORE 64 BASIC V2 ****" itd. oraz komunikat „READY." (czyli gotów). Mądrość interpretera jest jednak ograniczona i gdy wpiszemy mu „TY STARA BLASZANKO", to komputer odpowie komunikatem „?SYNTAX ERROR" (błąd składni) a nie np. „NO PROBLEMA, DURNIU", „HASTA LA VISTA, BABY" lub zwyczajnie „ODCZEP SIĘ, BARANIE". No dobrze, powiecie, ale o co mi właściwie chodzi? Już mówię.

Po prostu niniejszym przechodzimy do omawiania poleceń BASIC-a, które każdy z Was MUSI znać, jeśli chce cośkolwiek samemu zaprogramować. Niektóre z prostszych poleceń poznaliście już w poprzednich odcinkach (np. LOAD, SAVE). Dziś zajmiemy się również podstawowym ale i niezwykle ważnym poleceniem, o którym mówi się często, że jest „nieśmiertelne". I to prawda, bo bez niego nie powstałaby cała masa programów, więcej, istnienie komputerów straciłoby sens.

Cóż to za polecenie? Jest to **POKE**. Służy ono do wpisywania („wpoke'owywania") określonych wartości (liczb) do określonych komórek pamięci. Format POKE jest następujący:

POKE x,y

gdzie

x - adres komórki pamięci,
y - wartość (liczba) wpisywana do komórki.

Na wszelki wypadek nadmieniam, że w BASIC-u wszystkie wprowadzane liczby są dziesiętne (ale system dwójkowy trzeba znać, niedługo się przyda).

Parametry instrukcji POKE mogą być też wyrażeniami arytmetycznymi/algebraicznymi (ma to duże znaczenie podczas pisania programów), np.:

POKE 10000+7,5*4 albo
POKE Z, 4-A itp.

Znaczenie instrukcji POKE stanie się dla Was bardziej oczywiste jeśli Wam powiem, że komodorek ma pewne obszary pamięci, w których znajdują się specjalne rejestry odpowiedzialne za wyświetlanie obrazu na ekranie, za dźwięk, za pracę magnetofonu itp. Wystarczy do tych rejestrów wpisać odpowiednią wartość, a efekt będzie natychmiastowy. No i po to właśnie mamy POKE. Zobaczmy więc, jak to wygląda w praktyce. Wpiszmy:

POKE 53280,0 [RETURN]

Kolor ramki się zmienił, prawda? A teraz kolor tła:

POKE 53281,2 [RETURN]

Skutek natychmiastowy. Zróbmy jeszcze jedną rzecz. Wpiszmy:

POKE 1524,81

No i? Na środku ekranu pokazało się kółko. A teraz bardziej skomplikowany program:

```
10 FOR A=1024 TO 2023
20 POKE A,81
30 NEXT
40 GET A$:IF A$="" THEN 40
```

Wpiszcie te linie (po każdej naciskając [RETURN]) i następnie uruchomcie program poleceniem

RUN [RETURN]

Dobre, co? Cały ekran wypełnił się kółkami. Ostatnia linia programu powoduje, że komputer nie wyświetla od razu komunikatu „READY", lecz czeka na naciśnięcie dowolnego klawisza. Proponuję Wam pobawić się trochę tym programikiem (w linii 20 zamiast 81 możecie wpisywać dowolne liczby z zakresu 0 - 255).

Ale jak to zrobić? Naciśnijcie dowolny klawisz, wpiszcie LIST [RETURN]. Program ukaże się znowu na ekranie. Teraz naprowadźcie kursor na linię 20 i zmieńcie liczbę 81 na inną. Nie zapomnijcie zakończyć operację naciśnięciem klawisza RETURN. Teraz znowu wystarczy wpisać RUN [RETURN] - ekran wypełni się jakimś innym znakiem czy literą, zależnie od wprowadzonej zmiany.

I to wszystko na dziś, poeksperymentujcie sobie z instrukcją POKE a w następnym odcinku przejdziemy już do bardziej kompleksowego omawiania BASIC-a - przerobimy funkcje arytmetyczne, pętle i skoki. No to cześć, o Prześwietni!

ARNIE

> **Powinieneś wiedzieć, że**
>
> **POKE** - służy do wpisywania określonych wartości do określonych komórek pamięci.
> **LIST** - wyświetla na ekranie program znajdujący się aktualnie w pamięci komputera. Program taki można wówczas łatwo modyfikować.
> **RUN** - uruchamia program znajdujący się aktualnie w pamięci komputera.

10000000 - dziesiętnie 128
00101011 - " 43
11110000 - " 240
10001000 - " 136
01010101 - " 85



A MOŻE BY TAK MILIARDZIK?

Czyli o liczbach losowych słów kilka

Czy C-64 może generować idealne liczby losowe, tzn. takie, których pochodzenie jest rzeczywiście na sto procent przypadkowe? W jaki sposób można generować liczby losowe za pomocą komodorka? Czy można wykorzystać C-64 do gry w Lotto? Na te pytania spróbuję odpowiedzieć w tym artykule.

Pisząc coraz bardziej skomplikowane programy prędzej czy później każdy programista spotka się z problemem liczb losowych. Generacja liczb losowych jest potrzebna w grach, programach symulujących zjawiska fizyczne itp. W języku BASIC do uzyskiwania liczb losowych służy funkcja RND, która generuje liczby z przedziału 0-1. Jeżeli chcemy uzyskać liczby z innego zakresu, musimy posłużyć się wzorem **R=D+RND<0)*(G-D)**

gdzie wartość zmiennej G określa górną, a D - dolną granicę przedziału, z jakiego chcemy wylosować liczbę.

O ile w języku BASIC uzyskanie liczb losowych jest niezwykle proste, o tyle w kodzie maszynowym nastręcza to już więcej problemów. Teoretycznie można posłużyć się procedurą języka BASIC wywołując odpowiedni podprogram z pamięci ROM, jednakże częstokroć w naszych asemblerowych programach wyłączamy ROM. Można by to ominąć włączając i wyłączając go, lecz wykonanie tej operacji wymaga nie zmienionych wektorów BASIC-a i zostawia w pamięci ślad w postaci zmiennej, która może skasować fragment naszego programu.

Co więc zrobić? Najczęściej stosowane są dwie metody. Pierwsza polega na korzystaniu z generatora szumów ($d41b) SID-a. Przykładową procedurę pokazującą, jak należy to robić, obrazuje listing poniżej. Procedura INIT inicjuje generację liczb losowych i powinna być uruchomiona na początku programu. Procedura RND zwraca w akumulatorze wartość losową z przedziału 0-255 ($0-$ff).

```
;――――――――――――――――――――――――――――――――
;procedura generacji liczb losowych
;――――――――――――――――――――――――――――――――
init     lda #$ff
         sta $d40e
         sta $d40f
         sta $d413
         sta $d414
         lda #$8f
         sta $d418
         lda #$81
         sta $d412
         rts
;――――――――――――――――――――
rnd      lda$d41b
         rts
;――――――――――――――――――――
```

Jest to całkiem niezły sposób pod warunkiem, że nie użyje się SID-a do innych celów.

Druga metoda polega na odczytywaniu komórek pamięci, których wartość bardzo często ulega zmianie (np. rejest rastra - $d012). Jednak posługiwanie się tylko tą jedną komórką może okazać się zwodnicze, bo rzeczone $d012 przybiera pewne wartości częściej niż inne i na przykład prawdopodobieństwo uzyskania wartości z przedziału 0-54 ($00-$36) jest dwa razy większe niż prawdopodobieństwo uzyskania wartości z przedziału 55-255 ($37-$ff). Poza tym nie uzyskamy nigdy wartości odpowiadającym liniom, w których wykonuje się przerwanie graficzne.

Niedogodność tę można obejść dodając do siebie wartości odczytane z kilku komórek. Otrzymana w ten sposób wartość będzie „wystarczająco" losowa, bo jak wiadomo, błędy pochodzące z różnych źródeł kumulują się a nie znoszą. Dodatkowo za każdym razem możemy dodawać ostatnio otrzymaną wartość. Poniższa tabela przedstawia adresy niektórych komórek pamięci, które mogą posłużyć do generacji liczb losowych:

[illegible]

Adres	Funkcja
$d012	odczyt pozycji rastra
$d01e	rejestr kolizji sprajt-sprajt
$d01f	rejestr kolizji sprajt-tło
$d41b	oscylator głosu trzeciego/generator liczb losowych
$d41c	wyjście generatora obwiedni trzeciego głosu
$dc04	Timer A/CIA#1 młodszy bajt
$dc05	Timer A/CIA#1 starszy bajt
$dc06	Timer B/CIA#1 młodszy bajt
$dc07	Timer B/CIA#1 starszy bajt
$dd04	Timer A/CIA#2 młodszy bajt
$dd05	Timer A/CIA#2 starszy bajt
$dd06	Timer B/CIA#2 młodszy bajt
$dd07	Timer B/CIA#2 starszy bajt

Przykładowa procedura w Turboassemblerze 5.0 może wyglądać w ten sposób:

```
random1 lda $d012
        adc $dc04
        adc $dc05
        adc $dc06
        adc $dc07
        adc last
        sta last
        rts
;――――――――――――――――――――
last    .byte 00
```

No tak, ale w ten sposób można uzyskać tylko liczby z przedziału 0-255 ($00-$FF), a co zrobić jeżeli potrzebne są tylko liczby z mniejszego zakresu? Jeżeli górną granicą zakresu jest liczba X będąca potęgą dwójki pomniejszoną o jeden, to wystarczy dodać odpowiedni rozkaz AND #X przed rozkazem STA LAST. W przypadku gdy górną granicą naszego przedziału będzie liczba X nie będąca potęgą dwójki pomniejszoną o 1, należy wstawić przed rozkazem STA LAST rozkaz AND #Y, gdzie Y jest najmniejszą potęgą dwójki większą od X i pomniejszoną o jeden. Teraz trzeba jeszcze dopisać dodatkową procedurę, która będzie sprawdzać czy wylosowana liczba mieści się w odpowiednim zakresie i, w przypadku gdyby tak nie było, losuje ponownie:

```
losowa  jsr random2
        cmp #X+1
        bpl losowa
        rts
;――――――――――――――――――――
random2 lda $d012
        adc $dc04
        adc $dc05
        adc $dc06
        adc $dc07
        adc last
        and #Y
        sta last
        rts
;――――――――――――――――――――
last    .byte 00
```

Dobrze, a jeżeli potrzebujemy liczby z zakresu większego niż 0-255? Nic prostszego! Na przykład jeśli chcemy uzyskać liczbę z przedziału 0-65535, trzeba dwa razy wywołać procedurę RANDOM1 i pierwszy wynik potraktować jako starszy, a drugi jako młodszy bajt, lub odwrotnie.

Na zakończenie chciałbym podziękować Polonusowi/PADUA i Hainowi/ELYSIUM oraz Jarosławowi Kuśmierkowi, którzy swego czasu podsunęli mi kilka pomysłów. Czytelnikom życzę miłej zabawy z liczbami losowymi. Nie muszę chyba mówić, że wylosowane przez komputer liczby można użyć do gry w Lotto, być może uda się Wam wygrać miliardzik...

JETBOY/ELYSIUM



CZYM POTURBOWAĆ PROGRAM

Rozprawka o turbodoładowaniu C-64

Każdy, kto stał się szczęśliwym użytkownikiem wymarzonego C-64 z magnetofonem, po jakimś czasie stwierdza, że magnetofon jest dosyć kłopotliwą pamięcią masową. Na początku zauważa się wprost genialną szybkość standardowej transmisji danych z/do magnetofonu. Dzieje się to tak wolno, że aż niedobrze się robi. Czas ładowania programu ■ magnetofonu ■ normalnej transmisji jest taki sam jak w przypadku Atari 65XE z magnetofonem (bez turbo). W rzeczywistości komodorek wysyła/przyjmuje dane dwa razy szybciej, jednak dla poprawności zapisu i odczytu program zapisywany jest ■ następującej formie: nagłówek, informacje ■ programie (nazwa, długość), ponownie informacje o programie, ■ następnie dwukrotnie powtarzany jest cały blok danych programu. Korzyść wynika ■ tego taka, że mamy większą pewność, iż dane zostaną zapisane prawidłowo. Co jednak ■ niczym nie zmienia faktu, że czas potrzebny na zapis i odczyt programu wydłuża się dwukrotnie (a i tak jest wystarczająco długi).

Istnieje sposób, żeby skrócić sobie oczekiwanie. Jeśli Twój magnetofon ma wmontowany brzęczyk i słyszysz odczytywane dane, możesz wychwycić charakterystyczny sygnał dźwiękowy rozdielający oba bloki danych. Jeśli naciśniesz klawisz RUN/STOP ■ trakcie odczytu, po usłyszeniu tego sygnału możesz skrócić sobie ■ i od ■ uruchomić program.

Mimo stosowania takich niezbyt wygodnych metod czas ładowania programu ■ dalszym ciągu pozostawia wiele do życzenia. Ale programiści lubią ułatwiać sobie (i innym) życie i kiedyś ktoś wpadł na genialny pomysł, żeby wymyśleć coś takiego jak TURBO, czyli system przyspieszonej transmisji danych dla magnetofonu. Zapewne ■ dawna już go używasz. Początkowo były to krótkie programiki umożliwiające obsługę nowego systemu odczytu i zapisu. Obecnie większość ■ nas, użytkowników C-64 i magnetofonu używa takowego dopalacza zawartego ■ module (X, BLUE BOX, FINAL, ACTION). Wszystkie ■ nich używają systemu ■ nazwie TURBO ROM, jednak zanim ten standard ■ pełni się wykrystalizował, istniał jeszcze inny, który miał całkiem duże szanse na wyparcie TURBO ROM.

TURBO SAVE/LOAD

Tak właśnie toto się nazywało (w skrócie TSL). System ten był o tyle dobry, że program ■ zapisany ■ TSL nie wymagał ładowania programiku TURBO przed jego wczytaniem. TSL był osobnym programem, który pozwalał na załadowanie programu ■ normalnej transmisji i zapisanie go ■ tym systemie. Skracał znacznie czas transmisji danych.

Początek pliku TSL zapisywany był ■ normalnej transmisji - na początku znjdowała się procedura ładująca TURBO, która po załadowaniu (wszystko ■ normalnej transmisji) uruchamiała się i odczytywała właściwy program zapisany już w TURBO. System ten był ■ tyle niewygodny, że w przypadku ładowania programów zapisanych na różnych skosach głowicy nie można było się obejść bez dość kłopotliwego używania programu do ustawiania skosu głowicy. Także nagłówek zawierający całą procedurę ładującą znacznie wydłużał program wynikowy (zdarzało się, że odczyt procedury trwał dłużej niż odczyt samego programu). Programy zapisywane były bez opcji autorun (samouruchomienie). W końcu powstał więc inny, używany obecnie system:

TURBO ROM

Wersji tego systemu są dosłownie setki. Jedne lepsze inne gorsze. Różnią się one między sobą właściwie tylko kilkoma cechami: długością sygnału pilotującego, obsługą ■ widocznymi lub nie ■ trakcie odczytu i/lub zapisu paskami.

Program zapisany ■ TURBO ROM składa się przeważnie z krótkiego sygnału pilotującego, króciutkiego bloku zawierającego informacje o programie i głównego bloku zawierającego cały program. Wszystkie ■ elementy zapisane są w przyspieszonej transmisji, tak więc próby odczytu programu w TURBO ROM z poziomu „gołego” systemu operacyjnego kończą się komunikatem „?LOAD ERROR”. Wielką wadą tego systemu jest więc konieczność ładowania przed wczytaniem samego programu procedury ładującej.

Jak widać nie ma róży bez kolców. Kiedy postanawiasz przesiąść się na TURBO ROM, nagle trafiasz na duży problem, bo nie wiesz, który ■ loaderów wybrać. Potrzebny jest taki, który ■ trakcie odczytu obrazuje dane na ekranie ■ postaci pasków. Dlaczego? Na podstawie nagłówka programu (1, ■ sekundy równoległych poziomych pasków) można ustawić skos głowicy do danego programu. Należy dążyć do tego, żeby sygnał pilotujący ■ ekranie był zobrazowany ■ postaci idealnie równoległych pasków, przesuwających się płynnie po ekranie, bez żadnych dodatkowych zakłóceń. Wystarczy, że przećwiczysz to ■ kilku programach ■ będziesz wiedział ■ co chodzi.

Paski przy zapisie programu nie są wymagane. Jeśli chcesz zaoszczędzić miejsce na taśmie wybierz taki program, który generuje jak najkrótszy sygnał pilotujący. Jeszcze jedną, ważną cechą Twojego loadera powinno być to, aby nie blokował ■ możliwości zapisu i odczytu ■ normalnej transmisji. Jest to ważne o tyle, że czasami przypadnie Ci w udziale skopiowanie programu z normalnej transmisji na TURBO ROM i ■ odwrót.

Niektóre z programów obsługujących TURBO ROM kolidują ■ dłuższymi programami, uniemożliwiają np. zapisanie nazwy programu, bądź też powodują zawieszenie się programu. Trzeba więc odnaleźć taki loader, który nie będzie się tak brzydko zachowywał. Gdyby jeszcze ten loader podawał obszar pamięci, który zajmuje ładowany ■ TURBO ROM program, oraz upraszczał zapis określonego fragmentu pamięci, to wszystko byłoby wspaniale.

Przykładowao TURBO ABC używa do swoich celów komend LOAD i SAVE, co wyklucza możliwość używania standardowej transmisji. Sygnał pilotujący, zapisywany przez TURBO ABC jest także bardzo długi. Zdecydowanie nie polecam tego programu. Lepiej jest używać programu ■ nazwie TURBO ROM, który przy odczycie „wywala” na ekran paski, poza tym do jego obsługi używa się dwóch nowych komend (poprzedzonych strzałką ■ lewo): L (ładowanie) i S (zapis). Tak więc mamy dostęp do standardowej transmisji.



Magnetofon ■ najpopularniejsza ■ Polsce pamięć ■ dołączona do C-64. Problem tylko ■ tym, że jest ciut ■ wolna...

Czasem zdarza się, że TURBO ROM koliduje ■ odczytywanym programem i uniemożliwia jego uruchomienie. Poza tym w wypadku TURBO ROM ■ wielu wypadkach zdarza się, że wymagane jest zapisanie programu bez nazwy (nie pomaga nawet POKE 56,208). Takich problemów nie napotkacie np. używając TURBO zawartego ■ modułach FINAL, czy ACTION. W obydwu transmisja TURBO przyporządkowa■ jest urządzeniu logicznemu nr 7 i wszystkie operacje, jakie można przeprowadzać ■ przypadku transmisji standardowej są dostępne dla tego systemu.

Ale i ten ma swoje wady. Gdy chcesz np. zapisać tylko określony fragment pamięci, musisz albo określić początek i koniec programu ■ komórkach 43 - 44 i 45 - 46, albo też przejść do monito■ i stamtąd zapisać blok programu używając komendy: S"nazwa",07,xxxx,yyyy (xxxx - początek programu, yyyy - koniec programu, obie wartości zapisane ■ systemie szesnastkowym).

Pośród długotrwałych zabaw z najprzeróżniejszymi wersjami TURBO ROM znalazłem taką wersję, która zadowoli nawet najwybredniejszego użytkownika. Program ten, ■ nazwie HYPER TURBO, w trakcie odczytu pliku zapisanego w przyspieszonej transmisji obrazuje tę czynność paskami na ekranie, ■ zapisywany przez niego plik ma krótki nagłówek. Pozostawia także swobodny dostęp do standardowej transmisji. Co ważne - bardzo rzadko zdarza się, żeby program ten kolidował ■ jakimkolwiek innym. Udostępnia użytkownikowi cztery nowe polecenia (poprzedzone strzałką w lewo): L - odczyt programu, S - zapis, M - zapis programu z określeniem adresu początkowego i końcowego, Q - wyjście ■ programu (TURBO ROM zostaje wyłączony, co umożliwia uruchomienie niektórych problematycznych programów, po wyjściu ■ HYPER TURBO możliwe jest także ponowne uaktywnienie systemu). Dużym plusem tego programu jest to, że ■ wypadku dłuższych plików przeważnie nie ma kłopotów ■ zapisywaniem ich wraz ■ nazwą.

Czasami zdarza się, że Czytelnicy zadają nam kłopotliwe pytanie ■ programy kopiujące ■ taśmę. Właściwie odpowiedź powinna zabrzmieć: nie ma takich programów, bo nie są one potrzebne. Całej operacji kopiowania dokonuje się przy pomocy systemu operacyjnego i/lub programu TURBO ROM, ponieważ w większości wypadków programy nie uruchamiają się samoczynnie i bez większych problemów można je przed uruchomieniem zapisać.

BAD

NO MORE SKLEROSIS (MAGNETICUS)

Do redakcji „C&A" przychodzi bardzo wiele listów z prośbą, aby poruszyć temat zgrywania na nośnik określonych obszarów pamięci zarówno z poziomu języka BASIC jak i asemblera. Wychodząc naprzeciw Waszym oczekiwaniom napisałem ten oto artykuł zwracając uwagę, by zamieścić w nim możliwie dużo konkretnych przykładów. Mam nadzieję, że po takiej lekturze nie będziecie mieć więcej wątpliwości w tej kwestii.

Aby z poziomu BASIC-a zgrać określony obszar pamięci, należy wstawić adres początkowy (za pomocą instrukcji POKE) do komórek 43 i 44 (odpowiednio młodszy i starszy bajt), zaś do komórek 45 i 46 - adres końcowy, a następnie wykonać instrukcję:
SAVE"nazwa",8,1 (dla stacji dysków) lub
SAVE"nazwa",1,1 (dla magnetofonu).

Aby obliczyć starszy i młodszy bajt, wystarczy posłużyć się następującym wzorem:
SB=INT(WARTOŚĆ):MB=WARTOŚĆ-(SB*256)
gdzie:
WARTOŚĆ - wartość, którą chcemy przeliczyć na starszy i młodszy bajt,
SB - obliczany starszy bajt,
MB - obliczany młodszy bajt.
Aby nie było żadnych niejasności:
1. Obliczamy starszy i młodszy bajt adresu początkowego i końcowego (patrz powyżej).
2. Wykonujemy:
POKE 43,młodszy bajt adresu początkowego,
POKE 44,starszy bajt adresu początkowego,
POKE 45,młodszy bajt adresu końcowego,
POKE 46,starszy bajt adresu końcowego.
3. No i nagrywamy instrukcją:
SAVE"nazwa",8,1 - dla stacji dysków,
SAVE"nazwa",1,1 - dla magnetofonu.
Przykładowo zgrajmy sobie obszar pamięci od $c000 do $cfff:
POKE 43,0:POKE 44,192:POKE 45,255:POKE 46,207:SAVE"PROBA",8,1

Czyż nie proste? Tak, ale tego sposobu możemy używać tylko w trybie bezpośrednim. O tym, jak zgrać dany obszar pamięci w programie w BASIC-u napiszę za chwilę, zaś teraz zajmiemy się podobnymi procedurami w języku maszynowym. Oto przykład programu realizującego identyczne zadanie, jak podany przed chwilą ciąg instrukcji w BASIC-u:

```
.> c000 a9 06      lda #$06
.> c002 [illegible] [illegible]      ldx #$24
.> c004 [illegible] c0      ldy #$c0
.> c006 20 bd ff   jsr $ffbd
.> c009 a9 01      lda #$01
.> c00b a2 08      ldx #$08
.> c00d a0 00      ldy #$00
.> c00f 20 ba      jsr $ffba
.> c012 a9 c0      lda #$c0
.> c014 [illegible] fc      sta $fc
.> c016 a9 [illegible]      lda #$00
.> c018 [illegible] fb      sta $fb
.> c01a [illegible] [illegible]      lda #$fb
.> c01c [illegible] ff      ldx #$ff
.> c01e a0 cf      ldy #$cf
.> c020 20 [illegible] ff   jsr $ffd8
.> c023 60         rts
.:c024 50 52 4f [illegible] 41 2b 00 00 proba+..
```

Program uruchamia się poprzez SYS 49152. Na początku następuje ustalenie nazwy zgrywanego zbioru (nazwę należy wpisywać w kodzie ASCII):
lda #$06 - ilość liter w nazwie zbioru,
ldx #$24 - młodszy bajt adresu, w którym przechowywana jest nazwa zbioru,
ldy #$c0 - starszy bajt,
jsr $ffbd - wywołanie procedury SETNAM z KERNALa.

Następnie trzeba określić dodatkowe informacje, np. numer urządzenia, na które chcemy nagrywać:
lda #$01 - nr logiczny zbioru np. 1,
ldx #$08 - nr urządzenia, na które będziemy nagrywać (8 - stacja dysków, 1 - magnetofon),
ldy #$00 - adres wtórny 0,
jsr $ffba - wywołanie procedury SETLFS z KERNALa.

Na koniec należy określić adres początkowy i końcowy. Adres początkowy zapisać należy w dwóch sąsiadujących ze sobą, wolnych komórkach strony zerowej np. $FB $FC ($FB - młodszy bajt, $FC - starszy bajt):
lda #$c0
sta $fc
lda #$00
sta $fb - zapisanie w komórkach $fb i $fc adresu początkowego,
lda #$fb - wartość $fb ponieważ w $fb jest młodszy bajt adresu początkowego,
ldx #$ff - młodszy bajt adresu końcowego,
ldy #$cf - starszy bajt adresu końcowego,
jsr $ffd8 - wywołanie procedury SAVE z KERNALa,
rts - powrót do BASIC-a.

Pamiętajmy, że podczas zgrywania pamięci od adresu $a000 do $bfff komputer „widzi" tam ROM, a ściślej mówiąc - interpreter BASIC-a. Tak więc przed zgraniem należy go wyłączyć instrukcjami:

```
sei
lda $01
and #$fe
sta $01
```

Przed powrotem do BASIC-a interpreter należy z powrotem włączyć instrukcjami:

```
lda $01
[illegible] #$01
sta $01
cli
rts
```

Ładowanie dokładnie było omówione w „C&A" 7/92 w artykule „Autorun inaczej", zatem nie będę zagłębiał się w szczegóły tylko od razu przedstawię procedurę, która ładuje pierwszy zbiór z dyskietki i go uruchamia.

```
.> c000 [illegible] 01      lda #$01
.> c002 [illegible] 21      ldx #$21
.> c004 [illegible] c0      ldy #$c0
.> c006 [illegible] bd ff   jsr $ffbd
.> c009 a9 01      lda #$01
.> c00b [illegible] [illegible]      ldx #$08
.> c00d a0 01      ldy #$01
.> c00f [illegible] [illegible] ff   jsr $ffba
.> c012 [illegible] 00      lda #$00
.> c014 [illegible] ff      ldx #$ff
.> c016 a0 ff      ldy #$ff
.> c018 20 [illegible] ff   jsr $ffd5
.> c01b 20 59 a6   jsr $a659
.> c01e 4c [illegible] a7   jmp $a7ae

.:c021 [illegible] [illegible] [illegible] 00 00 [illegible] 00 00
*.......
```

Program uruchamia się poprzez SYS 49152. Dwa ostatnie skoki (jsr $a659 i jsr $a7ae) uruchamiają wgrany program (równoznaczne wpisaniu RUN).

Aby program wgrywał i uruchamiał pierwszy zbiór z taśmy, należy dokonać modyfikacji:

```
.> c000 a9 01      lda #$00
.> c00b [illegible] 08      ldx #$01
```

Pierwsza linia informuje komputer, że należy wgrać pierwszy napotkany program, druga zaś, że urządzeniem źródłowym ma być magnetofon.

Wróćmy jednak do sprawy zasadniczej, a mianowicie jak zgrać dany obszar pamięci w programie w BASIC-u. Proponuję skorzystać w tym celu z poniższego programu.

```
[illegible] D=49152:B=17212
205 C=0:E=D
210 READ A$:IF A$="END" [illegible] [illegible]
215 A1=ASC (LEFT$ (A$,1)) AND [illegible]
[illegible] A2=ASC (RIGHT$ (A$,1)) [illegible] [illegible]
225 IF A1>47 [illegible] 235
230 A1=A1+9:GOTO [illegible]
235 A1=A1-48
240 IF A2>47 [illegible] A2=A2-48:GOTO 250
245 A2=A2+9
250 A=A1*16+A2:POKE D,A
255 D=D+1:C=C+A:GOTO 210
[illegible] IF C<>B [illegible] PRINT „BLAD W [illegible]
    [illegible]A":[illegible]
265 PRINT CHR$ (147):PRINT „ZGRANIE PAMIECI:"
270 PRINT „SYS 49152,'NAZWA PLIKU',
    URZADZENIE,ADRES STARTOWY,KONCOWY"
275 PRINT:PRINT „ROUTINE [illegible] M.FERDYN"
[illegible] [illegible] 20,FD,AE,20,9E,AD,20,82
[illegible] DATA B7,8D,83,C0,A0,10,78,B1
290 DATA 22,99,84,C0,88,10,F8,58
[illegible] DATA 20,FD,AE,20,9E,B7,8E,82
300 DATA C0,20,FD,AE,20,8A,AD,20
305 DATA F7,B7,A5,14,8D,80,C0,A5
```

```
310 DATA 15,8D,81,C0,20,FD,AE,20
315 DATA 8A,AD,20,F7,B7,A5,14,8D
320 DATA 7E,C0,A5,15,8D,7F,C0,78
325 DATA A5,01,29,FE,85,01,AD,83
330 DATA C0,A2,84,A0,C0,20,BD,FF
335 DATA A9,01,AE,82,C0,A0,01,20
340 DATA BA,FF,AD,80,C0,85,FB,AD
345 DATA 81,C0,85,FC,A9,FB,AE,7E
350 DATA C0,AC,7F,C0,20,D8,FF,A5
355 DATA 01,09,01,85,01,60,00,00
360 DATA 00,00,00,00,00,END
```

Zmienia się nieco format instrukcji SYS 49152:
SYS 49152,"nazwa zbioru",urządzenie,adres początkowy,adres końcowy
gdzie
urządzenie: 1 - magnetofon, 8 lub 9 - stacja dysków, adres początkowy i końcowy - odpowiednio adres początkowy i końcowy obszaru pamięci przeznaczonego do nagrania (w systemie dziesiętnym).

Brak byłoby w tym artykule przysłowiowej kropki nad „i", gdybym nie przedstawił procedury pozwalającej ładować programy lub dane pod dowolny adres pamięci. Zadanie to wykonuje następujący program:

```
200 D=50000:B=13171
205 C=0:E=D
210 READ A$:IF A$="END" THEN 260
215 A1=ASC (LEFT$ (A$,1)) AND 63
220 A2=ASC (RIGHT$ (A$,1)) AND 63
225 IF A1>47 THEN 235
230 A1=A1+9:GOTO 240
235 A1=A1-48
240 IF A2>47 THEN A2=A2-48:GOTO 250
245 A2=A2+9
250 A=A1*16+A2:POKE D,A
255 D=D+1:C=C+A:GOTO 210
260 IF C<>B THEN PRINT „BLAD W LINIACH
    DATA":STOP
265 PRINT CHR$ (147):PRINT „LADOWANIE:"
270 PRINT „SYS 50000,'NAZWA',URZADZENIE,ADRES
    LAD."
275 PRINT:PRINT „ROUTINE BY M.FERDYN"
280 DATA 20,FD,AE,20,9E,AD,20,82
285 DATA B7,8D,B4,C3,A0,10,78,B1
290 DATA 22,99,B5,C3,88,10,F8,58
295 DATA 20,FD,AE,20,9E,B7,8E,B3
300 DATA C3,20,FD,AE,20,8A,AD,20
305 DATA F7,B7,A5,14,8D,B1,C3,A5
310 DATA 15,8D,B2,C3,78,A5,01,29
315 DATA FE,85,01,AD,B4,C3,A2,B5
320 DATA A0,C3,20,BD,FF,A9,01,AE
325 DATA B3,C3,A0,00,20,BA,FF,A9
330 DATA 00,AE,B1,C3,AC,B2,C3,20
335 DATA D5,FF,A5,01,09,01,85,01
340 DATA 60,00,00,00,00,00,END
```

Po uruchomieniu programu możemy wgrac nam dane względnie program pod wskazany adres. Format instrukcji uruchamiającej:
SYS 50000,"nazwa pliku",urządzenie,adres ładowania

Zamieszczone tu programy można stosować w swoich programach jednocześnie, tzn. zarówno ten do zgrywania, jak i drugi do wgrywania konkretnych obszarów pamięci. Przypomnę jeszcze, iż przy współpracy z magnetofonem program zostanie wczytany pod adres zapisany w nagłówku programu a nie pod ten, jaki podamy w instrukcji SYS 50000.

Mam nadzieję że artykuł ten wyjaśnił Ci, drogi Czytelniku, niektóre problemy i nigdy nie będziesz już miał kłopotów przy zapisywaniu bądź wczytywaniu danych, czego życzy Ci

MARIUSZ FERDYN

JAK Z KOMODORKA ZROBIĆ FILTROWNIĘ

W wielu listach docierających do redakcji powtarza się pytanie o przeznaczenie pinu 5 w gnieździe AUDIO/VIDEO. Otóż pin ten nosi nazwę AUDIO IN i służy do wprowadzania do C-64 sygnału dźwiękowego. Sygnał ten można następnie przetwarzać (filtrować) z wykorzystaniem SID-a (Sound Interface Device) czyli układu scalonego odpowiedzialnego za dźwięk. Do tego celu niezbędne jest jednak wykonanie odpowiedniego kabla łączącego wyjście magnetofonu z pinem 5 w gnieździe AUDIO/VIDEO. Sposób jego wykonania pokazany jest na rysunku. UWAGA!: W ŻADNYM WYPADKU NIE MOŻNA ZEWRZEĆ WYJŚCIA DŹWIĘKU Z MASĄ - jest to w 90% zgubne dla SID-a!. Natomiast trzeba pamiętać, żeby w magnetofonie stereofonicznym zewrzeć oba kanały (do C-64 musi dojść sygnał monofoniczny).

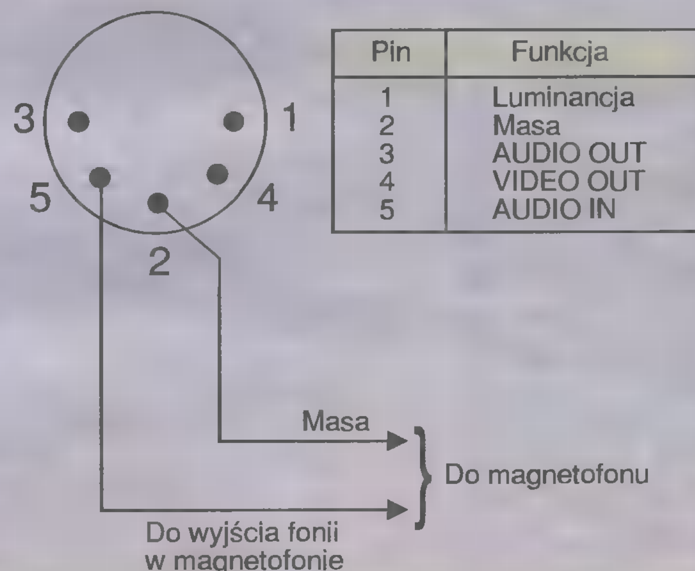
Po wykonaniu kabelka i spięciu nim magnetofonu z komodorkiem można przystąpić do eksperymentów muzycznych. Jednak aby w ogóle cokolwiek usłyszeć, trzeba ustalić poziom głośności dla generatorów dźwięku C-64. Odpowiada za to komórka 54296, a konkretnie jej młodszy nibble (cztery młodsze bity). Wynika z tego, że maksymalną wartością określającą głośność może być 15. Wpisujemy więc:
POKE 54296,15
i po tym magicznym poleceniu wokół rozlegnie się muzyka, o ile oczywiście macie w swym monitorze/telewizorze głośnik a głośność nie jest ustawiona na zero.

Teraz możemy zabrać się za filtrowanie nadchodzącego sygnału, ale przedtem trochę teorii. C-64 został wyposażony w cztery filtry:
- dolnoprzepustowy (wytłumia dźwięki o wysokich częstotliwościach),
- pasmowy (przepuszcza dźwięki o częstotliwościach zawartych w pewnym paśmie),
- górnoprzepustowy (wytłumia dźwięki o niskich częstotliwościach),
- rezonansowy (zwiększa - albo jak kto woli - podbija głośność dźwięków o zadanej częstoliwości).

Za filtrowanie sygnału doprowadzonego z zewnątrz odpowiedzialny jest bit 3 w komórce o adresie 54295, a więc zapalmy go:
POKE 54295,PEEK(54295)OR8
Filtr dolnoprzepustowy uaktywnia się zapalając bit 4 w komórce o adresie 54296:
POKE 54296,PEEK(54296)OR16
Filtr pasmowy załączamy za pomocą bitu 5 tej samej komórki czyli:
POKE 54296,PEEK(54296)OR32
Filtr rezonansowy załączamy za pomocą bitów 4-7 komórki 54295, które pozwalają na „podbicie" częstotliwości leżących blisko częstotliwości granicznej filtru. I to wszystko. Teraz możemy już pobawić się w filtrownię.

Opisanego tricku nie da się użyć do digitalizacji, gdyż nie jest to przetwarzanie analogowo-cyfrowe. Dźwięk, który wprowadzamy, nie jest wprowadzany do pamięci w postaci zrozumiałej dla naszego komodorka. W digitalizacje pobawimy się niebawem robiąc małą przeróbkę komodorowskiego magnetofonu.

MARCIN KULESZA/DRAGON



Pin	Funkcja
1	Luminancja
2	Masa
3	AUDIO OUT
4	VIDEO OUT
5	AUDIO IN

Baza Michasia

Pewien czas temu, a ściślej mówiąc w „C&A" 1/93, zaprezentowałem Wam pakiet edukacyjno-użytkowy „Groch z kapustą". Składało się nań kilka prostych programów doskonale spełniających swoje zadanie, a było nim przyjemne i bezbolesne wprowadzenie najmłodszych w świat komputerów. Pakiet ten wydała Fundacja Edukacji Technologicznej, która jest jedyną w Polsce firmą (i w dodatku taką z prawdziwego zdarzenia) zajmującą się produkcją/dystrybucją rodzimego oprogramowania dla Commodore 64. Niedawno, za sprawą FET-u na rynku ukazał się nowy program przeznaczony dla młodocianych użytkowników C-64 - „Baza Michasia".

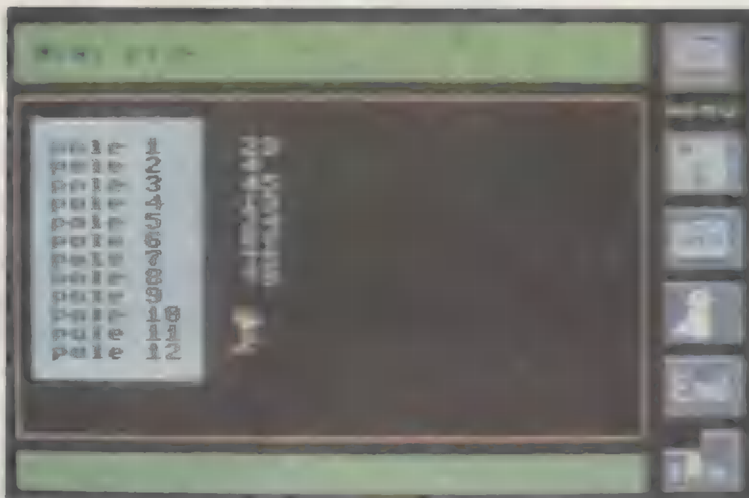
Co i w czym

Jest to najprawdziwsza baza danych, za pomocą której dosłownie każde dziecko z łatwością stworzy sobie podręczną kartotekę danych, książkę adresową, notatnik z telefonami kolegów itp. Również i starsi komodorowcy (zwłaszcza programiści) powinni się zapoznać z „Bazą Michasia", choćby po to, by zobaczyć, jak należy pisać programy przeznaczone dla milusińskich. Autorem programu jest znany już nam Pan Borys Żeszko.

Produkt otrzymałem do testowania bez żadnego opakowania, była to po prostu „goła" dyskietka (program rozprowadzany jest też na kasecie) z załączoną cienką instrukcją. Ta ostatnia zredagowana jest starannie, aczkolwiek od strony technicznej mogłaby być nieco lepsza (słaba jakość wydruku).

W działaniu

Już przy uruchamianiu programu natrafiłem na pierwszy problem. Otóż aby wczytać zasadniczą część programu należy podać hasło, którym jest wybrany losowo wyraz znajdujący się w instrukcji. Program niestety nie akceptuje wszystkich słów (poprawnie wpisanych), jakich on żąda. Czasem więc samo uruchomienie „Bazy Michasia" trwa parę minut, ponieważ dość często hasłem jest nie pierwszy czy drugi wyraz na danej stronie lecz np. dopiero piętnasty - zapewniam Was, że znalezienie go nie należy do ciekawych czynności. Jednak dzięki tej „szyfrowance" FET częściowo eliminuje nielegalne rozpowszechnianie „Bazy Michasia" przez piratów komputerowych, którzy - jak wiadomo - nie zawracają sobie głowy instrukcjami.

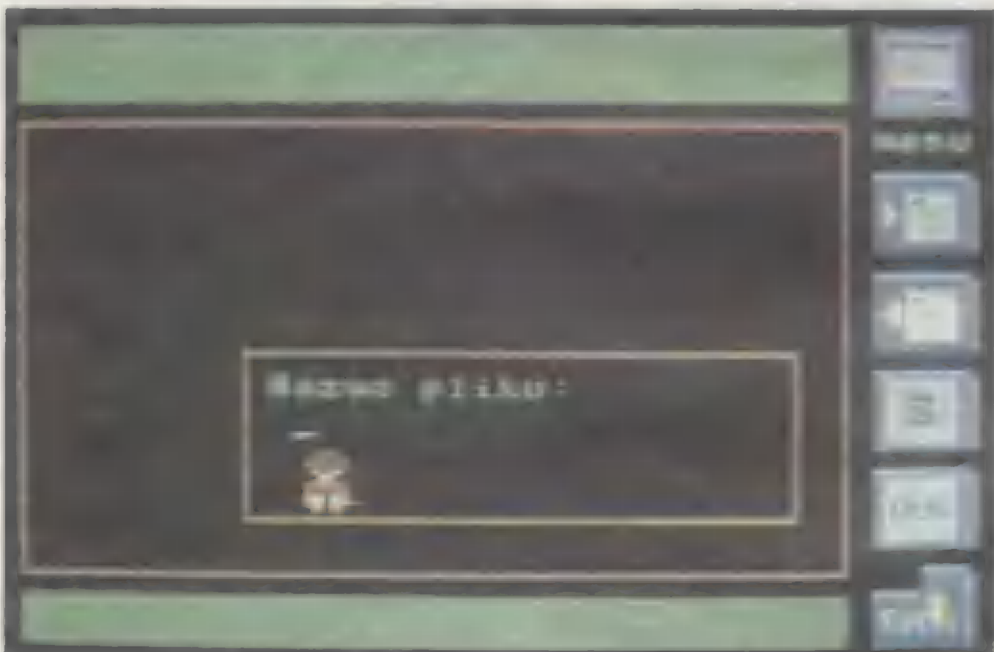

Podręczny arkusz kalkulacyjny

Po uruchomieniu programu oczom naszym ukazuje się menu główne. Z prawej strony ekranu znajdują się ikony reprezentujące dostępne opcje, z lewej zaś stoi Michaś - mały, śmieszny ludzik. Obsługa programu na tym etapie sprowadza się do naprowadzania kursora na żądaną ikonę i naciskania przycisku FIRE (następnie trzeba jeszcze potwierdzić wydane polecenie naciśnięciem ikony „ENTER"). Każdorazowo po wyborze jakiejś opcji do akcji wkracza Michaś - zaczyna poruszać się w kierunku otwartego aktualnie okna. Po dojściu do wyznaczonego miejsca na ekranie zatrzymuje się, a nad jego głową pojawia się kursor. Dopiero wówczas można wpisać odpowiednie dane i operacja zostaje zakończona.

„Wędrowki" Michasia trwają czasem dosyć długo zwłaszcza, jeśli miejsce, do którego ma dojść, znajduje się w przeciwległym rogu ekranu. Pamiętajmy jednak, że program został napisany z myślą o najmłodszych, a dla nich maszerujący ludzik stanowić będzie na pewno nie lada atrakcję.

Menu „Bazy Michasia" składa się aż z trzydziestu czterech ikon. Są one wystarczająco wyraźne, tzn. dobrze symbolizują czynność, z jaką są związane. W ogóle od strony graficznej program prezentuje się nieźle - poszczególne elementy współgrają ze sobą a całość jest gustowna i nie krzykliwa (a więc nie rozprasza niepotrzebnie użytkownika). Spodobał mi się również dobór barw, choć to już jest kwestią gustu. Autor wyposażył zresztą „Bazę Michasia" w opcję umożliwiającą zmiany koloru ekranu, ramek, liter i innych elementów obrazu.

Program współpracuje z drukarką, stacją dysków i magnetofonem. Samych poleceń kontrolujących pracę stacji dysków jest aż siedem. W czasie edycji danych mamy możliwość korzystania z jedenastu komend, a przy wydruku z czterech. Bez problemu można używać polskich liter, co jest ważne zważywszy na częściowo edukacyjny charakter programu.



Wybór pliku z dyskietki

A więc?

Z całą odpowiedzialnością mogę stwierdzić, iż „Baza Michasia" warta jest zakupu. Pozwoli ona młodemu użytkownikowi zrozumieć zasadę funkcjonowania baz danych a w przyszłości ułatwi „przesiadkę" na profesjonalne programy tego typu, jak choćby cBASE dla C-64 (dystrybutor: FET) a nawet dBASE czy FoxBase dla komputerów PC.

ROBERT KULIŚ

ZALETY:
+ wiele elementow edukacyjnych
+ przydatność
+ gustowna szata graficzna
+ duża liczba opcji (jak na program tego typu)

WADY:
- trudnosci podczas uruchamiania (złe odczytywanie haseł)
- nie całkiem profesjonalnie wydrukowana instrukcja

BAZA MICHASIA, dyskietka (lub kaseta) + instrukcja
AUTOR: BORYS ŻESZKO
DYSTRYBUTOR: Fundacja Edukacji Technologicznej, ul. Burdzińskiego 5, 03-480 Warszawa, tel. 180176

SOUND MONITOR

Podręcznik użytkownika (cz. 3)

Bardzo ważną sprawą w programie muzycznym jest możliwość zapisu nut do taktów z klawiatury w czasie realnym. Pozwala to zaoszczędzić wiele czasu. Równie istotna jest wygodna obsługa urządzeń zewnętrznych.

REALTIME RECORD-MODE

Jest to tryb zapisu taktów nutami granymi na klawiaturze komputera w czasie realnym - bardzo wygodna rzecz dla leniwych, jednak przed uruchomieniem tego trybu należy ustawić wiele parametrów dotyczących zapisu. Możliwe są dwie opcje: gra na klawiaturze z zapisywaniem lub bez. Druga z nich wymaga mniejszej liczby czynności i od niej zaczniemy.

Po inicjalizacji w każdym kanale i kroku zdefiniowany jest jeden głos (podstawa dźwięku) o numerze 00 - jest to tzw. taktomierz - przydaje się często do wybijania rytmu w czasie zapisu z klawiatury. Możemy go też użyć chwilowo jako głos prowadzący. Pierwszą czynnością w przygotowaniu do REALTIME RECORD-MODE (bez zapisu) jest ustalenie obszaru kroków (F3, F5), które będą użyczały swoich parametrów do REALTIME RECORD. Chodzi tu przede wszystkim o podstawę dźwięku, którą będziemy grać na klawiaturze. Ustalmy ten obszar na 2 pierwsze kroki (00-01). Dwie istotne tutaj możliwości SOUND MONITORa to zmiana oktawy na klawiaturze (klawisz „O" - w górę, SHIFT+"O" - w dół). Aktualna oktawa jest wyświetlona przy napisie „Octave" w środkowej części ekranu (normalnie C-0). Zmieńmy oktawę na C-5. Możliwa jest też transpozycja oktawy na klawiaturze o n półtonów w górę („K") lub w dół (SHIFT+"K").

Nie można zapomnieć o najważniejszym - należy wybrać kanały, na których będziemy grać. Można włączyć wszystkie kanały na raz, jednak w trybie zapisu zapisywany będzie i tak tylko jeden. Kanały 1, 2, 3 uaktywnia się/wyłącza odpowiednio klawiszami „,", „.", „/". Kanał włączony do grania sygnalizowany jest przez inwertowanie odpowiedniej cyfry przy „Play voice".

Możemy teraz uruchomić REALTIME RECORD klawiszem „F7". Kolor ramki zmieni się na czerwony. Klawisze, na których gramy, mają układ pianina, ułożone są w 2 rzędach - górny stanowią klawisze cyfrowe i górny rząd liter, dolny - środkowy i dolny rząd liter. W naszym wypadku CTRL jest dźwiękiem C, klawisz „1" - dźwiękiem C# itd.

Na naciśnięcie klawisza SID reaguje szumem dlatego, że na fali, którą „gramy", zdefiniowany jest szum (taktomierz). Powracamy do TRACK/STEP-TABLE naciskając [RETURN] lub „F7".

Zapis nut w takty wymaga trzech dodatkowych czynności. Trzeba ustalić, którą ścieżkę będziemy zapisywać przy pomocy klawisza „R" (bez SHIFT - w górę, z SHIFT - w dół). Nagłówek wybranego kanału zmienia kolor na żółty. Numer kanału wyznaczonego do zapisu widać także przy „Record track" (0 - kanał nie zdefiniowany). Jeżeli dokonaliśmy wyboru kanału, drugą ważną rzeczą jest nie tylko ustalenie początkowego i końcowego kroku zapisywanego, ale również wpisanie w tym obszarze (i wybranej ścieżce) niepowtarzających się taktów (np.: B000, B080, B0C0...).

Na koniec należy użyć komendy quantize (opcjonalnie) służącej do zaokrąglania dźwięku do wybranych długości. Quantize ustawia się klawiszami „Q" i SHIFT+"Q". Wartości quantize: 0 - quantize wyłączone, 1 - zaokrąglenie do szesnastki, 2 - do ósemki, 3 - do ćwierćnuty. Teraz wystarczy tylko nacisnąć „F7" i rozpocząć zapisywanie taktów.

Uwaga: zapisywanie rozpoczyna się z chwilą naciśnięcia „F7", trzeba więc uderzyć dźwięk w odpowiednim momencie. Z tą „synchronizacją" z programem mogą być pewne kłopoty. Ale i na to jest sposób. Wystarczy jeden kanał zająć metronomem, do kilkunastu kroków w tym kanale wprowadzić ten sam takt składający się np. z 2 ćwierćnut C-3. Potem ponowne włączenie zapisu i już słyszymy stukający taktomierz.

DISK-MODE

Tryb ten umożliwia współpracę SOUND MONITORa ze stacją dysków. Przed rozpoczęciem pracy z dyskietką należy wybrać klawiszem „T" typ pliku jaki chcemy wczytać lub zapisać. SOUND MONITOR tworzy i odczytuje trzy rodzaje plików:

Complete song (A000-CC00)

Jest to kompletny utwór razem z MUSIC MASTERem (procedura muzyczna). Przed zapisem takiego pliku należy pamiętać o uprzednim ustaleniu początku i końca utworu (F3, F5). Plik taki można wczytać osobno i uruchomić przez SYS 49152. Melodia będzie odtwarzana na przerwaniach (!).

Steps only (A000-C000)

Pliki tego typu są podobne do complete song z tym, że nie są „wyposażone" w procedurę muzyczną, ale za to są krótsze o ok. 4 KB, przez co oszczędzają miejsce na dyskietce.

Sounds

Ten typ plików jest wykorzystywany przy zapisie i odczycie (jednej lub więcej) podstaw dźwięku (podstawy dźwięku, które mają być odczytane lub zapisane, ustala się klawiszami „Y" - początek, „U" - koniec).

Po ustaleniu typu pliku używamy klawisza SHIFT+"L" (dla odczytu pliku) lub SHIFT+"S" (dla zapisu pliku). Kursor pojawi się na polu nazwy. Teraz wystarczy tylko wpisać nazwę pliku i gotowe (trochę wolno to działa, bo SOUND MONITOR nie ma żadnego „dopalacza"). Jeśli nie wpiszemy nazwy i naciśniemy [RETURN], program automatycznie wróci do TRACK/STEP-TABLE. Wpisując zamiast nazwy „$" możemy odczytać katalog dyskietki znajdującej się w stacji.

Graj, Cyganie, graj!

Po tak solidnej dawce teorii należy się Wam odpoczynek w postaci krótkiego utworu do wpisania. Oczywiście nie jest to żadne genialne dzieło, ale pokazuje jak w prosty sposób, przy użyciu tylko 4 taktów, napisać melodię.

Podział głosów w utworze powinien być mniej więcej taki: bas (najlepsza byłaby tu fala prostokątna, dobry efekt daje też ustawienie filtrów), melodia właściwa oraz akompaniament. Dodatkowo można zdefiniować perkusję, najlepiej na tym samym kanale co bas, ale z powodu braku 4 kanału pominę ten element (problem ten nie występuje w klonach SOUND MONITORa). W naszym wypadku bas będzie prowadził kanał 1, melodię - kanał 2, a akompaniament - kanał 3.

Na początku trzeba wpisać podstawy dźwięku, które będą wykorzystywane w utworze w trybie SOUNDS-MODE. Nie będziemy definiować podstawy dźwięku 00, bo jest ona zarezerwowana na metronom. Należy wpisać od góry do podstawy dźwięku 01 wartości (bas):

41,09,39,00,20,10,10,30,40,10,00,00,00,00,00,00,FF,00,00,
00,00,00,00,00;

podstawa dźwięku 02 (głos prowadzący):

21,3F,48,00,00,00,00,00,20,00,00,00,00,00,00,00,FF,00,00,
00,00,00,00,00;

podstawa dźwięku 03 (akompaniament):

11,08,70,00,20,10,10,00,10,00,00,00,00,00,00,00,FF,00,00,
00,00,00,00,00.

Po zdefiniowaniu podstaw dźwięku należy w TRACK/STEP-TABLE ustalić rozkład taktów w kolumnach, transpozycję i numery podstaw dźwięku dla taktów. Utwór będzie miał 8 kroków (od 00 do 07). Kroki te powinny wyglądać w następujący sposób:

[illegible]	TRK1	TR	[illegible]	[illegible]	TR	[illegible]	[illegible]	[illegible]	[illegible]	ARP/S
[illegible]	B000	[illegible]	01	BE00	[illegible]	[illegible]	BE00	[illegible]	00	BF00
[illegible]	B000	[illegible]	01	BE00	00	[illegible]	BE00	00	[illegible]	[illegible]
[illegible]	B000	FE	01	BE00	00	[illegible]	[illegible]	[illegible]	[illegible]	BF00

```
03 [illegible] 00 01 BE00 [illegible] [illegible] BE00 [illegible] [illegible] BF00
[illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] 00 [illegible] BF00
[illegible] [illegible] [illegible] 01 [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible]
[illegible] [illegible] [illegible] 01 [illegible] [illegible] 02 [illegible] [illegible] [illegible] [illegible]
07 [illegible] [illegible] 01 [illegible] [illegible] 02 B080 [illegible] [illegible] BF00
```

Gotowe! Teraz należy tylko wpisać takty B000, B040, B080. Takt B000 (nuty powinny być w rewersie - podawane bez SHIFT) określa nuty dla basu:

```
[illegible] C-2 [illegible] -- [illegible] -- 00 -- [illegible]
B008 E-2 00 -- [illegible] -- 00 -- 00
B010 C-3 00 -- 00 -- [illegible] -- [illegible]
B018 E-3 [illegible] -- [illegible] -- [illegible] -- [illegible]
B020 C-2 [illegible] -- 00 -- [illegible] -- [illegible]
B028 E-3 00 -- 00 -- [illegible] -- [illegible]
B030 C-4 00 -- [illegible] -- [illegible] -- [illegible]
B038 E-4 [illegible] -- [illegible] -- [illegible] -- [illegible]
```

Takt B040 - nuty melodii:

```
B040 C-4 00 +++ [illegible] +++ [illegible] +++ [illegible]
[illegible] +++ [illegible] +++ [illegible] -- [illegible] -- [illegible]
B050 E-4 [illegible] +++ [illegible] +++ [illegible] +++ 00
B058 +++ [illegible] +++ 00 -- [illegible] -- 00
B060 E-4 [illegible] +++ 00 +++ 00 -- 00
[illegible] F-4 [illegible] +++ 00 +++ 00 -- 00
B070 E-4 [illegible] +++ [illegible] +++ 00 -- [illegible]
B078 G-4 00 +++ [illegible] +++ 00 -- 00
```

Takt B080 - akompaniament:

```
B080 E-4 [illegible] -- [illegible] -- [illegible] -- [illegible]
[illegible] -- 00 -- 00 -- [illegible] -- 00
B090 -- 00 -- 00 -- [illegible] -- [illegible]
B098 -- [illegible] -- 00 -- 00 -- [illegible]
[illegible] E-4 00 -- 00 -- 00 -- 00
B0A8 -- 00 -- [illegible] -- [illegible] -- [illegible]
B0B0 -- 00 -- [illegible] -- [illegible] -- 00
[illegible] -- [illegible] -- [illegible] -- [illegible] -- 00
```

W ten sposób przy pomocy trzech pełnych i jednego pustego taktu, 3 podstaw dźwięku i przy użyciu transpozycji stworzyliśmy krótki utwór. Po ustaleniu jego początku (krok 00-F3) i konca (krok 07-F5), możemy go posłuchać naciskając klawisz „N”.

Praca z magnetofonem

Dla tych, którzy nie posiadają stacji dysków ale mają do swojego C-64 jakąś przystawkę (X, FINAL II/III, EXPERT), kilka słów na temat pracy SOUND MONITORa z magnetofonem (obsługa na przykładzie FINAL II).

Po wczytaniu programu należy uruchomić go bez KILL, nacisnąc RESET, wpisać KILL i wrócić do SOUND MONITORa (SYS 4096). Po stworzeniu utworu można go nagrać np. w TURBO ROM. Należy zresetować SOUND MONITOR, wejść w tryb monitora FINALa (SHIFT+"F1") i wpisać:

.S"nazwa utworu",07,A000,CC00 (odpowiada „complete song").

Można także nagrać utwór bez MUSIC MASTERa:

.S"nazwa utworu",07,A000,C000 (odpowiada plikowi „steps only").

W celu zapisania podstaw dźwięku trzeba wpisać:

.S"sound",07,AE00,xxxx (odpowiada plikowi „sounds"),

gdzie xxxx oznacza adres $AE00 + $18 x liczba podstaw dźwięku. Pliki te wczytuje się spod FINALa przez LOAD"",7. Po wczytaniu należy wykonać: RESET, KILL, SYS 4096.

[illegible] i mutancie

Na koniec chciałbym jeszcze napisać kilka słów na temat jednej z mutacji SOUND MONITORa - DRUM MAKERa, jako że na pewno wielu z Was dysponuje właśnie tym programem. Nawiasem mowiąc powstał on na bazie gry TO BE ON TOP.

DRUM MAKER ma nieco przyjemniejszy wygląd od SOUND MONITORa. Zmieniony został zestaw znaków. Dolny lewy róg ekranu zajmują wskaźniki dynamiki poszczególnych kanałów (od góry: 1, 2, 3 głos, perkusja).

Do DISK-MODE przechodzi się naciskając „L" lub „S" bez SHIFT. W trybie tym możliwe jest używanie 6 rodzajów plików:

1. DRUMS ONLY (9000-A000) - tylko układ perkusji.
2. SOUNDS (AE00-B000) - podstawy dźwięku.
3. STEPS ONLY (A000-C000) - muzyka bez digitalizacji i [illegible] MUSIC MASTERa.
4. DRUM+MUSIC (6000-CD00) - kompletny utwór (perkusja+muzyka).
5. DIGI DATES (obszar w zakresie 4000-9000, w zależności od [illegible] pliku, najczęściej 6000-9000) - zbior danych dźwięku digitalizowanego - tylko odczyt.
6. KONVERT SM (A000-CC00) - odpowiada plikowi „complete song" SOUND MONITORa, konwersja danych z SOUND MONITORa.

Takt pusty w DRUM MAKERze jest oznaczony jako B800. Istnieje tylko jedno ARP/S pod adresem BF00; nie można ustawić parametrów podstawowych i arpeggio dla każdego kroku. Zamiast tego w kolumnie ARP/S mamy perkusję. Wartość 9000 oznacza pusty takt perkusji. Takty perkusji wpisujemy od 9020 w dół. Takt perkusji składa się z 4 linii po 8 bajtów (standardowo). Każda z linii jest początkowo (po inicjalizacji) wypełniona zerami (00 - brak dźwięku). W miejsca te wpisuje się numer dźwięku digitalizowanego określonego parametrami w TIME/BEGIN-STOP.

Zmieniła się także inicjalizacja pamięci - zadawane są trzy pytania o inicjalizację perkusji (drums), kroków i taktów (music) oraz podstaw dźwięku (parameters).

Do tabeli TIME/BEGIN-STOP przechodzi się z TRACK/STEP-TABLE naciskając „X". Wygląda ona tak samo jak TRACK/STEP-TABLE z tym, że funkcjonuje tylko okno systemowe. Zmieniło się znaczenie kolumn - kolumny oznaczające w TRACK/STEP-TABLE wskazania na takty dla głosu 1, 2, 3 - tutaj kolejno: TIME (czas trwania dźwięku digitalizowanego), BEGIN (początkowy adres tego dźwięku), STOP (końcowy adres). Maksymalna długość takiego dźwięku - 4 KB. Możliwy do zajęcia przez dane do odtwarzania digitalu obszar - $4000-$9000. Odtwarzanie jednego dźwięku w TIME/BEGIN-STOP uzyskuje się przez naciśnięcie [RETURN] na tym dźwięku. W TIME/BEGIN-STOP ustalone są następujące dźwięki:

01 - gitara basowa,
02-0F - perkusja,
11 - perkusja,
18-1E - synteza mowy,
20-2B - gitara basowa,
30-37 - gitara basowa,
38 - synteza mowy,
40-4B - gitara basowa (dubluje 20-2B),
50-57 - gitara basowa,
58-5F - synteza mowy,
60-6B - gitara basowa,
70-77 - gitara basowa (częściowo dubluje 60-6B).

DRUM MAKER ma jednak dwie poważne wady: niemożność kopiowania taktów oraz brak możliwości wykorzystania REALTIME RECORD-MODE do zapisu nut w takty.

I to już naprawdę wszystko. Przypominam, że SOUND MONITOR jest dostępny na naszej dyskietce PD nr 16. Życzę wszystkim miłego komponowania.

BARTŁOMIEJ DRAMCZYK

MULTIFASTPLOT

Nasi stali czytelnicy zapewne pamiętają, że w „C&A" 2/92 publikowaliśmy procedurę PLOT, mogliby więc zapytać, dlaczego poruszamy ten temat po raz drugi. Jednakże poprzednia procedura dotyczyła trybu wysokiej rozdzielczości (320x200), ten natomiast obsługuje tryb graficzny, w którym mamy do dyspozycji 4 kolory w rozdzielczości 160x200 pikseli.

Procedura jest napisana w turboasemblerze, tak że łatwo można ją dołączać do własnych programów, ponadto jest ona w pełni relokowalna. Osoby pragnące użyć ją w swoich programach muszą zadbać o to, aby procedura i program nie wykorzystywały tych samych komórek pamięci na stronie zerowej. Mogłoby to w efekcie dać różne, nieprzewidziane skutki. A oto jak należy wywoływać odpowiednie funkcje procedury:

FASTPLOT - przed wywołaniem do rejestru A podajemy numer koloru z zakresu 0-3, do rejestru X podajemy współrzędną x (od 0 do 159) a do rejestru Y - współrzędną y (od 0 do 199). Wywołanie FASTPLOT-a ze źle podanymi wartościami może spowodować zawieszenie komputera i utratę zawartości pamięci.

INITPLOT - przed wywołaniem do rejestru Y podajemy starszy bajt adresu, od którego zaczyna się mapa bitowa ekranu. Ta procedura powinna być wywołana przed użyciem FASTPLOT-a lub FCLEAR.

FCLEAR - procedura szybkiego kasowania zawartości ekranu.

Program nie należy do krótkich, dlatego nie publikujemy go w piśmie (brak miejsca i chętnych do wklepywania tak długich listingów) - jest dostępny na naszej dyskietce PD nr 17.

RAFAŁ PIASEK

DAGMARA EDYTOR

Dagmara Edytor to program służący do projektowania duszków. Nadszedł do naszej redakcji w liście od Autora, Kamila Prejsa, jako materiał do rozpowszechniania w konwencji Public Domain (niezorientowanym przypominam, że program rozprowadzany jako PD kazdy może sobie skopiować, poprawić, dowolnie zmienic, a nawet - UWAGA UWAGA! - wykorzystać jego fragmenty we własnym, komercyjnym programie). Program ten nie jest szczytem informatycznego



geniuszu, ale swoje funkcje - to jest projektowanie sprajtów i nagrywanie ich na dyskietkę - spełnia w stopniu całkowicie zadowalającym. Do tego ma jeszcze kilka całkiem pożytecznych funkcji, o których później.

Edytor Kamila Prejsa jest zbyt długi, by można go było wydrukować w naszym piśmie. Wg moich pobieżnych szacunków, zająłby ok. 5 stron „C&A", co byłoby zbyt ciężką próbą cierpliwości dla Czytelników, nawet niekoniecznie tych, którzy chcieliby wpisać ów program, także dla całej reszty, która szuka tu innych, ciekawych rzeczy. Rozważywszy to wszystko zdecydowaliśmy się umieścić program na czerwcowej dyskietce „C&A" (dysk PD nr 17), którą można kupić wpłacając 30.000 zł na konto Spółdzielni „Bajtek". Szczegóły, jak zwykle, na stronie 34.

Program, jak już wspominałem, służy do projektowania sprajtów, zarówno jednobarwnych, jak i wielokolorowych. Oto krótka instrukcja obsługi - mam nadzieję, że będzie ona pomocna dla wszystkich, którzy zdecydują się skorzystać z tego narzędzia.

Klawiszologia

Kursorem kierujemy za pomocą joysticka podłączonego do portu nr 2. Naciskając przycisk strzału zapalamy punkt pod kursorem, gasimy go zaś za pomocą spacji. A tak działają inne klawisze:

- **C** czyści całą matrycę aktualnie oglądanego sprajta,
- **+/-** zmienia edytowanego sprajta,
- **SHIFT+/SHIFT-** też zmienia obrabiane sprajty, ale znacznie szybciej,
- **</>** zmienia kolor sprajta w trybie jedno barwnym,
- **F7** przełącza tryb wielokolorowy,
- **1/2/3** pozwalają wybrać zmienić kolor, którym aktualnie rysujemy,
- **SHIFT 1/2/3** zmieniają kolory w trybie wielobarwnym,
- **F2** pokazuje duszka nr 00 (od adresu $4000),
- **F8** pokazuje duszka nr ff (od adresu $7fc0),
- **F3** rozszerza lub zwęża sprajta w poziomie,
- **F5** rozszerza lub zwęża sprajta w pionie,
- **S** symetria duszka w osi pionowej,
- **M** symetria duszka w osi poziomej,
- **P** pozwala przesuwać wzór duszka w matrycy za pomocą klawiszy kursora (wyjście - RETURN),
- **O** włącza/wyłącza układ współrzędnych (jak w „okrętach" - pionowo - litery, poziomo - cyfry),
- **J** daje możliwość zmiany prędkości kursora (1 - najszybciej, 7 - najwolniej, 3 - chyba najwygodniej),
- **(funt)** zapamiętuje wzór duszka w buforze roboczym,
- **(strzałka w górę)** wstawia do aktualnej matrycy wzór duszka z bufora roboczego,
- **H** pokazuje szybką instrukcję obsługi,
- **Q** wyjście z programu.

Operacje dyskowe

- **D** powoduje wyświetlenie katalogu dyskietki,
- **L** wczytuje wzory duszków nagrane na dysku,
- **S** nagrywa sprajty zaznaczone za pomocą opcji M.

Operacje na zaznaczonych duszkach

- **M** oznacza duszka lub zdejmuje oznaczenie,
- **SHIFT/M** zaznacza grupę duszków (od duszka xx do duszka yy),
- **N** kasuje wszystkie oznaczenia,
- **@** zamienia miejscami oznaczonego duszka z edytowanym duszkiem,
- **A** pokazuje animację zaznaczonych sprajtów.

Warto wiedzieć

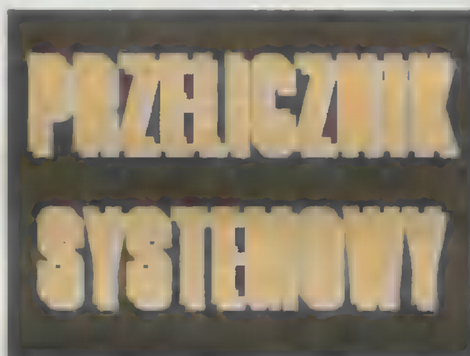
Żeby program nie zawieszał się kiedy to nie jest niezbędne, należy wyłączyć moduły: FINAL (instrukcja KILL), ACTION REPLAY (ZAP), BLACK BOX, X (wyłącznik na obudowie) i inne (odpowiednio).

Jeśli program mimo wszystko się zawiesi należy zresetować komputer i nakazać wykonanie instrukcji SYS 4096.

Wiele zabawy z duszkami... NIECH VIC BĘDZIE Z WAMI!

ANTENI NADAWSKI

PS. Serdecznie dziękujemy autorowi Kamilowi Prejsowi za nadesłanie tego programu. Zapraszamy również innych Czytelników do współpracy z „C&A".



Prezentowany program umożliwia przeliczanie liczb pomiędzy trzema systemami: dwójkowym, dziesiętnym i szesnastkowym. Autorem programu jest Mariusz Maciąg z Kętrzyna.

Po załadowaniu i uruchomieniu programu wybieramy system, w którym wpiszemy liczbę do przeliczenia. Do wyboru mamy:

b - system binarny, czyli dwójkowy,
d - system dziesiętny,
h - system szesnastkowy (heksadecymalny).

Po dokonaniu wyboru naciskamy RETURN i wpisujemy liczbę. Ponowne naciśnięcie klawisza RETURN spowoduje przeliczenie podanej liczby na dwa pozostałe systemy. Powtórzenie operacji uzyskujemy poprzez naciśnięcie dowolnego klawisza.

ROBERT KULIŚ

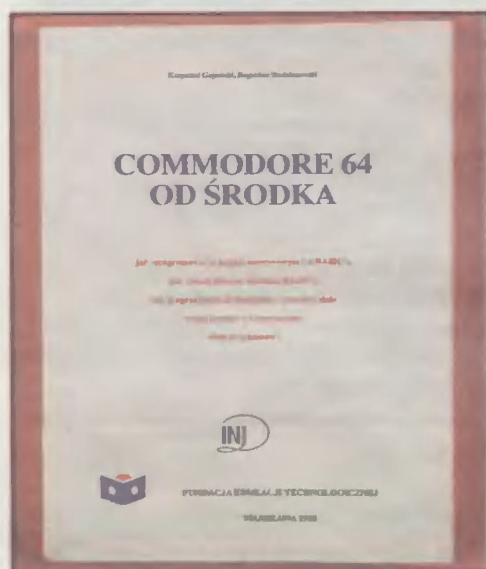
PS. Serdecznie dziękujemy autorowi za nadesłanie programu. Jednocześnie informujemy, że nie będziemy więcej (przez dłuższy okres czasu) publikować programów tego typu - ostatnimi czasy zamieściliśmy ich kilka, to chyba wystarczy.

```
5 clr:input"Podaj system
  (b=1,d=2,h=3) ";a
10 on a goto 15,40,65:goto 5
15 clr:input"Bin ";a$
20 z=49:gosub 235:if g=1 then 15
25 f=2:gosub 90:gosub 275
30 f=16:gosub 140:gosub 280
35 goto 225
40 clr:input"Dec ";a$
45 z=57:gosub 235:if g=1 then 40
50 f=2:gosub 140:gosub 270
55 f=16:gosub 140:gosub 280
60 goto 225
65 clr:input"Hex ";a$
70 z=57:x=65:y=70:gosub 235:if
   g=1 then [illegible]
75 f=16:gosub 90:gosub 275
80 f=2:gosub 140:gosub 270
85 goto 225
90 for s=1 to l
95 c$=mid$(a$,s,1)
100 c=val(c$)
105 if c$="a" then c=10
110 if c$="b" then c=11
115 if c$="c" then c=12
120 if c$="d" then c=13
125 if c$="e" then c=14
130 if c$="f" then c=15
135 w=w+c*f^(l-s):
   next:a$=str$(w):return
140 a=val(a$):c$=""
145 b=int(a/f)
150 c=a-b*f
155 if b=0 then 180
160 if c=0 then 170
165 gosub 180:goto 175
170 c$="0"+c$
175 a=b:goto 145
180 gosub 185:c$=m$+c$:return
185 m$=right$(str$(c),1)
190 if c=10 then m$="a"
195 if c=11 then m$="b"
200 if c=12 then m$="c"
205 if c=13 then m$="d"
210 if c=14 then m$="e"
215 if c=15 then m$="f"
220 return
225 get y$:if y$="" then 225
230 goto 5
235 l=len(a$):if a$="" then [illegible]
240 for s=1 to l
245 h=asc(mid$(a$,s,1))
250 if h>=x and h<=y then [illegible]
255 if h>=48 and h<=z then 265
260 g=1:return
265 next:return
270 print:print"Bin   "c$:return
275 print:print"Dec  "w:return
280 print:print"Hex   "c$:return
```

Commodore 64 od środka

Dawno, dawno temu, kiedy po ziemi łaziły jeszcze dinozaury, kiedy ludzie posługiwali się krzesiwem by rozniecić ogień i nie znali nawet procedury FLI, panowie Krzysztof Gajewski i Bogusław Radziszewski opublikowali książeczkę „Jak rozbudować interpreter?". Natychmiast rzuciły się na nią sfory posiadaczy C-64, dzięki czemu łączny nakład paru wydań przekroczył 100000 egzemplarzy, co może w porównaniu do różnych „Harlekinów" nie jest liczbą oszałamiającą, lecz i tak dużą.

W książce tej Autorzy zamierzali zapewne rozpropagować swoje największe osiągnięcie - dialekt WARSAW BASIC. Na kilkudziesięciu stronach objaśnili, jakie są ogólne zasady działania interpretera BASIC-a w C-64 oraz jak ów interpreter można poszerzyć o własne polecenia i instrukcje. W kolejnych rozdziałach cierpliwie tłumaczyli co dzieje się, gdy komputer otrzyma jakieś polecenie w BASIC-u i jak zmusić go, by robił to, co chcemy, na czym polega programowanie proceduralne i strukturalne oraz obsługa błędów. Jako przykład rozszerzenia BASIC-a wydrukowali listingi programu WARSAW WEDGE (uproszczona wersja WARSAW BASIC-a). Wszystko to zajęło Autorom 96 stron.



Przyczyna, dla której jestem entuzjastą tej książki zaczyna się od strony 97. Tu bowiem zamieszczono pełny wydruk pamięci ROM, żeby było ciekawiej - z komentarzami wpisywanymi z iście benedyktyńską cierpliwością przy prawie każdym wierszu. W książce tej wylistowano cały ROM - od $a000 do $bfff i od $c000 do $ffff. Kopalnia gotowych procedur przygotowanych przez projektantów C-64. Są tu podprogramy do nagrywania i odczytywania, drukowania, operacji na liczbach - wiele bardzo pożytecznych rzeczy, których znajomość ułatwia pisanie wszelkich programów. Możliwość wglądu do pamięci ROM w każdej chwili dość często się przydaje, bowiem znacznie łatwiej jest napisać jedno „JSR" niż kilkadziesiąt innych rozkazów asemblera.

Stosunkowo niedawno panowie Autorzy przypomnieli sobie o tej książce i postanowili powtórzyć jej sukces. Nakładem Fundacji Edukacji Technologicznej i Wydawnictwa IND wydali książkę „Commodore 64 od środka". Błyszcząca okładka, biały papier (a nie żółty albo szary), wyraźny druk - znaki czasu, które już na pierwszy rzut oka odróżniają tę książkę od mojego starego, wyświechtanego egzemplarza „Jak rozbudować interpreter?". Podczas pobieżnego kartkowania zauważyłem charakterystyczny układ strony, właściwy dla komentowanych wydruków pamięci. Tak! To był wydruk ROM-u. Treściowo książka ta różni się od poprzedniej jedynie dodaniem kilku rozdziałów na temat nauki programowania w asemblerze, co uważam zresztą za duży plus publikacji.

„Commodore 64 od środka" to książka, która stanowić powinna żelazną pozycję w podręcznej biblioteczce każdego, kto poważnie myśli o programowaniu. Zwłaszcza z powodu wydruku zawartości ROM-u. W tej bowiem materii publikacja ta jest JEDYNĄ znaną mi pozycją na polskim rynku, co też powinno zwiększyć jej popularność. Naprawdę porządny kawał dobrej roboty pp. Gajewskiego i Radziszewskiego.

BARTŁOMIEJ KACHNIARZ

„Commodore 64 od środka", str. 262
Autorzy: Krzysztof Gajewski, Bogusław Radziszewski
Wydawcy: Fundacja Edukacji Technologicznej i Wydawnictwo IND

Tytuł / Liczba kolejnych zeszytów	3	6	12	liczba egz.
Bajtek	X	75000	150000	
C&A	30000	60000	X	
TOP SECRET	37500	75000	X	

Kupon ważny do 30.06.1993 r.

Co by zaprenumerować...

Bajtek – Magazyn komputerowy dla wszystkich - początkujących i zaawansowanych, dużych i małych, 8- i 16- bitowych.

C&A – Miesięcznik dla posiadaczy C-64 i Amig - programowanie, używanie, kabelki, stacje, czyli wszystkiego po trochu.

TOP SECRET – Supermagazyn o grach nie wymagający specjalnego reklamowania.

Warunki prenumeraty:

- Prenumerata zawarta przed upływem ważności kuponu gwarantuje stałość cen.
- Przesyłka pocztowa nie wymaga dodatkowych opłat.
- Jeżeli w ciągu 2 tyg. od pojawienia się numeru w kioskach przesyłka nie nadeszła, prosimy o kontakt.
- Za błędy wynikające z niestarannego wypełnienia formularza redakcja nie ponosi odpowiedzialności.
- Prosimy o staranne i wyraźne wpisanie odpowiednich liczb egzemplarzy.
- Na kopercie z kuponem prosimy wyraźnie napisać "PRENUMERATA".

NINIEJSZYM ZAMAWIAM:

Dysk(i) PD na AMIGĘ:
nr ____________________

Dysk(i) PD na C-64/128:
nr ____________________

Kaseta na C-64 nr 1 2 3 4

IMIĘ: ....
NAZWISKO: ....
DOKŁADNY ADRES: ....
.... **(KOD)** ....

KUPON WAŻNY DO 10. 07. 1993

Ceny: dyski (AMIGA, C-64) – 30.000 zł/szt.
kaseta (C-64) – 100.000 zł/szt.

Pieniądze prosimy wpłacać na konto:
Spółdzielnia „Bajtek"
Bank „AGROBANK" S.A.,
Warszawa, ul. Grochowska 262,
konto nr 470005-1834-131
STARANNIE I CZYTELNIE wypełniony kupon
z dowodem wpłaty należy przysłać na adres:
Spółdzielnia „Bajtek"
ul. Raperswilska 12, 03-956 Warszawa
UWAGA! Kupon jest opatrzony datą ważności.
Zamówienia dokonywane na kuponach,
które utraciły ważność (decyduje data stempla pocztowego)
nie będą realizowane.

AMIGA - dysk PD nr 13

Rend24
Jest to program do konwertowania pojedyńczych obrazków lub ich zestawów na obrazki i animacje w formacie umożliwiającym wyświetlenie ich na Amidze. Program rozpoznaje i przekształca następujące formaty: 24-bit ILBM, GIF i JPEG. Umożliwia zmianę formatu, liczby kolorów oraz posiada opcję ditheringu. Otrzymane obrazki można zapisać w dowolnym formacie dostępnym na Amidze.

Ham Lab Demo
Podobnie jak Rend24 służy do przekształcania obcych formatów na „dające się obejrzeć" formaty Amigi. Różni się on od poprzednika liczbą rozpoznawanych formatów (w tym wiele używanych przez programy do ray-tracingu). Ma również więcej opcji pozwalających ditherować obrazki według różnych algorytmów.

Kurve
Jest to program do wyświetlania i drukowania funkcji matematycznych podanych w formie analitycznej. Wykreśla pierwszą i drugą pochodną danej funkcji oraz oblicza całkę oznaczoną w podanych granicach. Podaje również miejsca zerowe i ekstrema funkcji.

Cross Maze
W kąciku gracza kolejna gra umysłowa. Z zestawu słów o różnej długości należy „wypełnić" krzyżówkę. Gra przypomina nieco typową krzyżówkową „Jolkę" z tą różnicą, że nie trzeba odgadywać haseł (i całe szczęściem bowiem program jest w wersji angielskiej).

C-64 - dysk PD nr 17

Oprócz programów zamieszczonych w tym numerze „C&A" na dyskietce znajduje się rewelacyjne demo grupy CREST pt. „Avangarde". Zajmuje ono całą stronę dyskietki, dlatego nic więcej nie udało nam się zmieścić. Demo jest jednak naprawdę super i warto je obejrzeć.

TO JEST TO!

UWAGA, UWAGA! Wszyscy posiadacze magnetofonów, oto propozycja nie do odrzucenia! Wydajemy dwie nowe kasety z programami na C-64.
KASETA NR 3 - zawiera programy, które ukazały się na łamach "C&A" w numerach 1-6/1993.
KASETA NR 4 - SUPERPROPOZYCJA - nagrana dwustronnie, a więc całe 60 minut programów typu Public Domain. Oferta zaiste godna uwagi, gdyż wśród 50 pozycji znalazły się ZNANE DEMA, INTRA, PROGRAMY MUZYCZNE oraz SAMPLINGI. Słowem naprawdę ekstra grafika i muzyka razem do kupy wzięte.
Pragniemy poinformować Was także, że w ciągłej sprzedaży są dwie poprzednie kasety:
KASETA NR 1 - zawierająca programy z "C&A" 2-6/1992.
KASETA NR 2 - zawierająca programy z "C&A" 7-12/1992.
Cena kaset nie zmieniła się i nadal wynosi 100000 zł/szt.
Wszystkim, którzy do tej pory nabyli nasze taśmy i dyskietki serdecznie dziękujemy, życząc udanych i pogodnych wakacji.

REDAKCJA

Zamierzam kupić monitor kolorowy do Amigi 500+. Proszę o powiedzenie mi jaki mam wybrać po możliwej cenie.

Michał

Proponuję na początek zastanowić się, czy naprawdę warto i czy jest Ci potrzebny monitor kolorowy. Może zamierzasz np. używać Amigi do edycji tekstów? W takim przypadku kupowanie kolorowego monitora jest bez sensu.
Zaś jeśli już zdecydowałeś na zakup kolorowego monitora, to mogę tu śmiało polecić dwa dość popularne na naszym rynku modele. Są to: Commodore 1084S i PHILIPS CM8833/II. Bez problemu uzyskasz na nich obrazy w najwyższej rozdzielczości dostępnej na Amidze, a ich parametry techniczne można określić jako bardzo dobre. Ceny tych monitorów wahają się od 3,5 do 4,5 mln zł.

OAK

◆

Mam zamiar kupić stację dysków 3,5" i 5,25" i tu wynika problem: do Amigi nie można chyba podłączyć dwóch stacji dysków, ponieważ jest tylko jedno gniazdo dla stacji dysków?

Daniel

Do Amigi można podłączyć aż trzy zewnętrzne stacje dysków. Będą one miały symbole df1:, df2:, df3:. Dołącza się je w taki sposób, że pierwszą stację podłączamy do gniazda w komputerze, a kolejne do gniazd w stacjach dysków. Jednak niektóre stacje nie są wyposażone w takie gniazdo. Wówczas trzeba dorobić je samemu, co wcale nie jest takie trudne.

DUD

◆

Jestem mało doświadczonym posiadaczem A500. Proszę o odpowiedź na nurtujące mnie pytania:
1. Od czego i w jaki sposób powstaje napis „Guru meditation" po zawieszeniu się gry?
2. Bardzo często po wgraniu gry ukazuje się ramka, w której widnieje napis „Error validating disk". Potem program prosi o wgranie Disk Doctora z Workbencha. Jednak rezultatów nie ma.

Mariusz

Ad. 1. Napis „Guru Meditation" jest efektem błędu w programie lub nierespektowania przez niego zasad współpracy z systemem operacyjnym. Ramka Guru pokazuje numer błędu oraz adres, pod którym on wystąpił. Aby się dowiedzieć, co dany numer błędu oznacza, trzeba skorzystać z programu Guru Info lub innego podobnego.
Ad. 2. Ten napis oznacza nieuporządkowanie mapy zajętości dysku, np. po operacji zapisu. Może być efektem zbyt wczesnego wyjęcia dyskietki z napędu lub np. działalności wirusa. W założeniu problem miał rozwiązywać dodawany przez Commodore program Disk Doctor, jednak tylko pogarsza on sytuację. Radziłbym używać raczej programów QuarterBackTools lub FixDisk.

DUD

◆

Czy do Amigi 500 lub 600 można podłączyć monitor kolorowy Commodore 1802D?

Krzysztof

Tak, jest to możliwe, z tym, że do A500 konieczny jest modulator, aby uzyskać kolorowy obraz. W przypadku A600 możliwe jest połączenie bezpośrednie.

DUD

◆

Mam problem, w numerze 4/92 „C&A" znajduje się schemat samplera do samodzielnego montażu. W układzie scalonym z wyjścia CLOCK wychodzi przewód do dwóch inwerterów I1 i I2, tam wszystko się urywa. Przecież układ inwerterów nie może wisieć w powietrzu.

Dominik

Po pierwsze: CLOCK jest wejściem, a nie wyjściem układu scalonego. Na to wejście podaje się sygnał zegarowy taktujący przetwornik ADC0808/9.
Po drugie: Sygnał ten jest wytwarzany przez generator zbudowany z inwerterów I1 i I2. Nic tu nie wisi w powietrzu. Sygnał z wyjścia układu generatora jest podawany przez kondensator na wejście, ulega wzmocnieniu w inwerterze I1 i jest odwracany w I2, po czym cały proces jest powtarzany. Sygnał prostokątny, powstający w efekcie działania układu jest podawany na wejście CLOCK.

DUD

◆

Od kilku miesięcy posiadam stację dysków 1541-II do Commodore C-64. (...) Zakupiłem ją od kolegi wraz z modułem Action Replay Plus V7.3. Mam kilka problemów, a mianowicie:
1. Większość programów i gier na dyskietkach nie wczytuje się. Ładują się dobrze, ale tylko na początku, później zaczyna migotać dioda DRIVE, stacja przez chwilę „buczy" i następuje ?LOAD ERROR.
2. Zakupiłem dyskietki PRECISION, z których tylko pierwszy lub dwa pierwsze programy wczytują się a z resztą jest tak, jak napisałem powyżej. Kupiłem też VERBATIM, ale te okazały się jeszcze gorsze. Natomiast z dyskietek STILON, NARWATRONIC i KODAK, które to otrzymałem przy zakupie, gry wczytują się prawidłowo (chociaż nie wszystkie).
3. Przy przegrywaniu gier z taśmy postępuję zgodnie z instrukcją. Załadowują się niby dobrze, ale później się nie wczytują z dyskietek. Moje dyskietki: PRECISION 2S-HC, VERBATIM Verex MD2-HD, STILON 2S/2D 48 TPI, KODAK MD2-D.
Czym to wszystko jest spowodowane?! Czy trzeba kupować inne dyski, czy też stacja nie funkcjonuje prawidłowo? Zaznaczam, że stacja była czyszczona!

Przemysław

Sądzę, że Twój problem bierze się z wyeksploatowania stacji. Podczas długiego użytkowania, parametry techniczne głowicy zapisująco-odczytującej nieco się zmieniają i pojawiają się problemy z odczytaniem co starszych dysków. Poza tym trochę groźnie i trochę dziwnie brzmi nazwa procesu „czyszczenia stacji". Nie wiem, na czym właściwie ma on polegać, ale może on spowodować niemożność odczytywania dysków poprzednio na tej stacji zapisanej - znowu mogą się pojawić jakieś zmiany w charakterystyce głowicy. Nota bene znam przypadek człowieka, który czyścił głowicę zwykłego adapteru. Jakiś paproch bardzo mocno się przylepił, więc trzeba było dość energicznych zabiegów by go usunąć. Po pewnym czasie okazało się, że adapter stracił swój odczytujący diamencik - ten właśnie paproch.
A co do dyskietek - nie kupuj dyskietek oznaczonych jako „gęste" (HD). Kupuj tylko dyski podwójnej gęstości - 2D albo DD (double density). Kupowanie dyskietek high density to strata pieniędzy - zawsze pojawiają się na nich błędy.

BAK

◆

Nazywam się Grzegorz Skorupa (...) mam komputer COMMODORE 64II + magnetofon (...).
1. Chciałbym się dowiedzieć, jak się tworzy grafikę i plansze tytułowe w języku maszynowym? Skąd można wziąć kod maszynowy?
2. Czy jest jakiś wzór na zamianę liczby dziesiętnej na szesnastkową i odwrotnie?
3. W jaki sposób (jak, gdzie) można zakupić programy oryginalnych firm np. DOMARK, BEAU-JOLLY zaprezentowane na łamach „C&A"?

Serdeczne pozdrowienia

Grzegorz

1. Ciężko powiedzieć w kilku zdaniach. Potrzebny jest do tego dobry program graficzny (np. Art Studio), joystick, dużo wolnego czasu i talent graficzny. Natomiast muszę się przyznać, że nie rozumiem pytania „Skąd wziąć kod...?". Najlepiej z głowy, ewentualnie z pomocą programów typu MONITOR lub ASEMBLER i kursów programowania opublikowanych w naszym piśmie.
2. Jasne, że jest! Radzę zajrzeć do dziewiątego odcinka przygód z armatą.
3. Radzę dokładnie czytać publikowane przez nas recenzje gier. Zawsze drukujemy wraz z nimi adres polskiego dystrybutora. O szczegóły dotyczące sprzedaży radzę zwrócić się właśnie do niego.

BAK

◆

Nazywam się Marek Obuchowicz, mam 9 lat (...). Mam kilka problemów (...).
1. Final II: Po wciśnięciu FREEZE ukazuje się menu:
F1=Menu 2 F3=Dsave F5=Tsave F7=Print
Po wybraniu F3 pojawia się napis:
RESET DRIVE OR START TAPE RETURN
Wciskam REC+PLAY, wciskam RETURN i C-64 nagrywa program na DYSK!
2. Przyślijcie mi kody maszynowe typu: a9 (HEX)=LDA, 141(DEC)=STA
3. W grze Pit Fighter (i nie tylko) po przejściu przez level 1 komputer chce odczytać level 2 z dysku (?!). Przyślijcie mi zestaw POKE'ów, aby komputer odczytywał kolejny level z taśmy.

Marek

1. W menu Final II Dsave to skrót od „Disk Save", a Tsave - od „Tape Save". Jeśli więc chcesz nagrać na taśmie, to zamiast F3 naciskaj F5.
2. Nie przyślemy. Możemy najwyżej odesłać Cię do literatury: „Procesor 6502 i jego rodzina", „Asembler 6502" Jana Ruszczyca oraz innych książek traktujących o kodzie maszynowym. Podobną rozpiskę opublikujemy na zakończenie kursu asemblera.
3. Nie przyślemy. Kupujesz gry od piratów i dziwisz się, że nie chcą Ci działać poprawnie. Krajowi piraci jakiś czas temu wypuścili mnóstwo gier nagrywanych przez moduły typu Final lub Action. WSZYSTKIE miały do siebie to, że biesiły się po utracie wszystkich (lub nawet jednego!) żyć albo po przejściu poziomu.

BAK

◆

Bardzo zależy mi na odpowiedzi na to pytanie: co to jest CP/M? Czy 64-ka może pracować w tym systemie? A jeśli nie, to jak ją można do tego systemu przystosować?

Rafał

CP/M to system operacyjny, dyskowy. Był popularny i rozpowszechniony w latach siedemdziesiątych i osiemdziesiątych. Był jednym z pierwszych systemów operacyjnych stosowanych w komputerach domowych. Możnego protektora znalazł w firmie Amstrad, która standardowo implementowała go w swoich maszynach (664, 6128, 8256). W pewnym momencie firma Commodore specjalnie zainstalowała w C-128 procesor Z80 po to głównie, by można było korzystać z CP/M-u. Niektórzy stosowali go z różnym powodzeniem w ZX Spectrum. Prawdziwi, rasowi maniacy instalowali go w C-64 łącznie z procesorem Z80. Ale system ten zaczął stopniowo zanikać i już między połową a końcem lat osiemdziesiątych zajmowali się nim jedynie miłośnicy zabytków. CP/M pozostawił jednak trwały ślad - jego elementy zostały wykorzystane podczas tworzenia systemu MS-DOS i AmigaDOS.

BAK

ATARI FALCON 030 CONTRA AMIGA 1200

Widok z okopów Atari

W marcowym numerze „Commodore & Amiga" ukazał się artykuł porównujący komputery Atari Falcon 030 i Amigę 1200. Pomimo szczerych chęci autora, w tekście znalazło się sporo nieścisłości, które utworzyły gęstą mgłę utrudniającą obserwację pola walki. Proponuję więc rozwiać ją co nieco.

Konstrukcja

Falcon został zbudowany w oparciu o wydajny, 32-bitowy mikroprocesor Motorola 68030 taktowany częstotliwością 16 MHz. Maksymalna moc obliczeniowa komputera wynosi 3,84 MIPS-ów (bez procesora sygnałowego). Zainstalowana w A1200 Motorola 68EC020 popędzana zegarkiem 14,18 MHz osiąga tylko 2,35 MIPS-a. W Falconie umieszczono podstawkę pod koprocesor matematyczny MC68881 lub MC68882, wystarczy dokupić układ scalony i wstawić w odpowiednie miejsce. W przypadku A1200 trzeba SAMEMU (!) wlutować lub zlecić firmie (koszty!) zainstalowanie podstawki (w montażu powierzchniowym) pod koprocesor. Już widzę kolejki do punktów serwisowych Commodore. Drugim wyjściem jest nabycie dodatkowej karty z wmontowaną MC 68882 i dodatkowymi 2 MB FAST-RAM-u. Jednak koszt takiego wynalazku działa hamująco na większość użytkowników Amigi - cena giełdowa 4,5 do 6 milionów złotych.

W Falconie dostęp procesora do zasobów sprzętowych komputera jest możliwy po magistarali systemowej lub poprzez kanały DMA. Te ostatnie mogą być tak konfigurowane, aby dane mogły być przenoszone w dowolnych kierunkach, procesor wizyjny - pamięć operacyjna, stacja dysków - DSP itp. Motorola 68030 posiada pamięć podręczną typu cache podzieloną na dwa obszary: dane (256 bajtów) i rozkazy (256 bajtów). W przypadku braku dostępu do magistarli procesor i tak zajęty jest realizacją programu zawartego w cache'u. MC 68EC020 nie ma cache'u na dane. Amiga 1200 dostępna w sprzedży ma zainstalowane tylko 2 MB CHIP-RAM-u. Tak więc nie jest możliwy jednoczesny dostęp procesora i układów graficznych do pamięci operacyjnej. I znowu, zwiększenie szybkości graficznej A1200 przez dodanie choćby 16-bitowego FAST-RAM-u na kartach PCM CIA 2.0 kosztuje dosyć drogo.

Dodatkowo Falcon ma wbudowany rodzaj pamięci nieulotnej tzw. NV RAM, którą można traktować jak CMOS RAM w pececie. W odróżnieniu od „blaszaka" NV RAM nie jest podtrzymywana bateryjnie (prawdopodobnie zbudowano ją w oparciu EEPROM). Mieszczą się w niej dane o konfiguracji systemu, używanej wersji językowej OS i klawiatury, czasu itp.

System operacyjny

Według sugestii autora wymienionego na wstępie artykułu, system operacyjny Atari oparty został o przestarzały CP/M 68K. Należy się tu przynajmniej parę słów wyjaśnienia. Pod koniec 1984 roku, kiedy prototyp Atari ST został już uruchomiony, firma Atari zwróciła się do firm software'owych z propozycją napisania systemu operacyjnego dla nowego komputera. Początkowo zakceptowano CP/M 68000, który był rozwinięciem systemu CP/M pierwotnie stworzonego dla mikrokomputerów opartych o procesor INTEL 8080, poźniej ZILOG Z-80. Jednak w wyniku pewnych trudności zrezygnowano z CP/M 68000, ale od tej samej firmy zakupiono GEM czyli graficzny system zarzadzania zasobami sprzętowymi komputera. Resztę oprogramowania napisała już firma Atari. Nowy system nazwano TOS. W pierwszy modelach ST musiał być on wczytywany z dyskietki. Jednak kolejne wersje instalowano już w pamięci ROM. Atari Falcon ma zainstalowany w ROM-ie TOS o numerze 4.01 (obecnie 4.03) i nie jest on wielozadaniowy. Aby pobawić się w multitasking wystarczy wczytać z dyskietki MultiTOS, dostarczany z każdym Sokołem.

Grafika

Programowany procesor graficzny Falcona umożliwia pracę w różnych rozdzielczościach graficznych zależnych w zasadzie od jakości podłączonego wyświetlacza. Oczywiście najlepszym jest tu monitor typu Multisync. Przy korzystaniu z desktopu Atari (odpowiednik amigowskiego Workbencha) można używać następujących rozdzielczości:

ST Low - 320 na 200, 16 kolorów
ST Med - 640 na 200, 8 kolorów
ST High - 640 na 400, 2 kolory
oraz
Falcon Low - 320x200, 65536 kolorów max
Falcon Mid - 320x480, 65536 kolorów max
Falcon High - 640x480, 65536 kolorów max
Falcon SuperHigh - 1280x960, 65535 kolorów max

Ostatni tryb jest dostępny w Falconach, które zeszły z taśm produkcyjnych po kwietniu 1993 roku. System wizyjny nowego Atari może byc synchronizowany z zewnątrz, a pobrany tą drogą obraz może być utrwalony na magnetowidzie przy pomocy... modulatora telewizyjnego. Jeżeli jakość obrazu nam nie wystarcza, możemy zakupić za niespełna 100 DM Genlock, który przetwarza sygnały wizyjne z jakością akceptowalną w studiu telewizyjnym.

Muzyka

System dźwiękowy Falcona zasługuje na większą uwagę. Dwa 16-bitowe przetworniki cyfrowo-analogowe umożliwiają samplowanie i odtwarzanie dźwięku o jakości odtwarzacza płyt CD. Dla zapewnienia kompatybilności z poprzednimi modelami STE wbudowano dodatkowo dwa 8-bitowe przetworniki cyfrowo-analogowe oraz znany z ST generator dźwięku Yamahy. Obecny na pokładzie DSP (Motorola 56001/32 MHz/16 MIPS) umożliwia symulowanie 8 kanałów dźwiękowych oraz tworzenie w czasie rzeczywistym takich efektów jak Chorus, Flanger, Pogłos, Equalizer itp. Sam fakt powstania CUBASE AUDIO na Falcona jest miarą wysokiej jakości wbudowanych układów dźwiękowych. Do tej pory CUBASE AUDIO dostępny był jedynie razem z bardzo drogą Yamahą CBX D5 (ale o tym innym razem). Dla porównania karta muzyczna do Amigi 2000/3000/4000 o symbolu AD516 (firmy SunRize) o zbliżonych możliwościach do Falcona kosztuje 1495 $. Jak zapowiadają szefowie Commodore, dodatkowe układy muzyczne do A1200 mają kosztować ponad 500 DM. Jednak jak dotąd nie pojawiły się na rynku komputerowym.



Podsumowanie

Na koniec warto przeprowadzić prosty rachunek. Atari Falcon 030 kosztuje 12 mln zł. (z 1 MB RAM-u), zaś Amiga 1200 9 mln i dodatkowe przetworniki 5 mln zł. W obu przypadkach kwota wynosi około 13 mln zł. Atutem Amigi jest 2 MB RAM-u, zaś Falcona - Motorola 68030, prosta instalacja kooprocesora matematycznego, DSP 56001, dźwięk i wbudowany napęd dysków elastycznych 1,44 MB.

Tak to wygląda, moi mili, z okopów Atari. Ende.

TAJNY KURYER „C&A"
Robert MM Chojecki

Od redakcji:
Aby uniknąć krwawych sporów między atarowcami i amigowcami spieszymy z wyjaśnieniem, że z racji szczupłości miejsca temat został potraktowany dosyć powierzchownie. Jeżeli jednak strony będą domagały się pojedynku, zobowiązujemy się do zorganizowania ringu - obszernego artykułu porównującego oba komputery.

